

Bazy danych - dokumentacja projektu

Michał Piwowarczyk

January 2024

Spis treści

1	Założenia i funkcjonalności	3
2	SQL	3
2.1	tabele.sql	3
2.2	widoki_triggery.sql	3
2.3	inserty.sql	3
2.4	selecty.sql	3
3	Tabele	3
4	Przepływ danych	4
5	Relacje między tabelami	4
6	Atrybuty tabel	5
6.1	pilkarz	5
6.2	trener	5
6.3	fizjoterapeuta	5
6.4	sala	6
6.5	sprzet	6
6.6	trening	6
6.7	cwiczenia_fizjoterapeuta	6
6.8	pilkarz_trening	7
7	Analiza zależności funkcyjnych i normalizacja tabel	7
8	Triggery	7
8.1	Sprawdzanie pozycji	7
8.2	Usuwanie zależnych	9
8.2.1	Usuwanie pilkarza	9
8.2.2	Usuwanie trenera	9
8.2.3	Usuwanie fizjoterapeuta	10
8.2.4	Usuwanie treningu	10
8.2.5	Usuwanie sali	11

8.2.6	Usuwanie sprzętu	11
9	Widoki	11
9.1	Widok treningu	11
9.2	Widok ćwiczeń fizjoterapeutycznych	12
9.3	Tygodniowe wydatki na treningi	13
9.4	Tygodniowe wydatki na fizjoterapeutów	13
9.5	Tygodniowe wydatki na trenerów	13
9.6	Tygodniowe wydatki na sale	13
9.7	Tygodniowe wydatki na sprzęt	14
9.8	Tygodniowe wydatki na fizjoterapeutów	14
10	Aplikacja	14
10.1	Strona startowa	15
10.2	Budżet	16
10.3	Wyświetl dane	17
10.4	Plan	18
10.5	Plan osoby	19
10.6	Plan ogólny	20
10.7	Admin	21
10.8	Dodaj	22
10.9	Dodaj piłkarza	23
10.10	Dodaj trening	24
10.11	Przypisz piłkarza do treningu	25
10.12	Dodaj ćwiczenia fizjoterapeutyczne	26
10.13	Usuń	27
11	Makropolecenia	28
12	Wprowadzanie danych	28
13	Baza danych	28

1 Założenia i funkcjonalności

Tematem projektu jest stworzenie bazy danych wraz z interfejsem graficznym służącej do zarządzania klubem piłkarskim. Klub dzięki niej ma możliwość kontrolowania wydatków oraz zarządzania treningami i ćwiczeniami. Baza danych ma umożliwiać przeglądanie rekordów w tabelach w niej zawartych oraz usuwanie i dodawanie rekordów do tabel.

2 SQL

W projekcie znajdują się 4 pliki sql.

2.1 tabele.sql

Ten plik odpowiada za tworzenie tabel.

2.2 widoki_triggery.sql

W tym pliku znajdują się wszystkie stworzone widoki oraz triggery.

2.3 inserty.sql

W tym pliku znajdują się inserty odpowiadające za wprowadzenie przykładowych danych do tabel

2.4 selecty.sql

Tu znajdują się wszystkie zapytania jakie zostały użyte w projekcie.

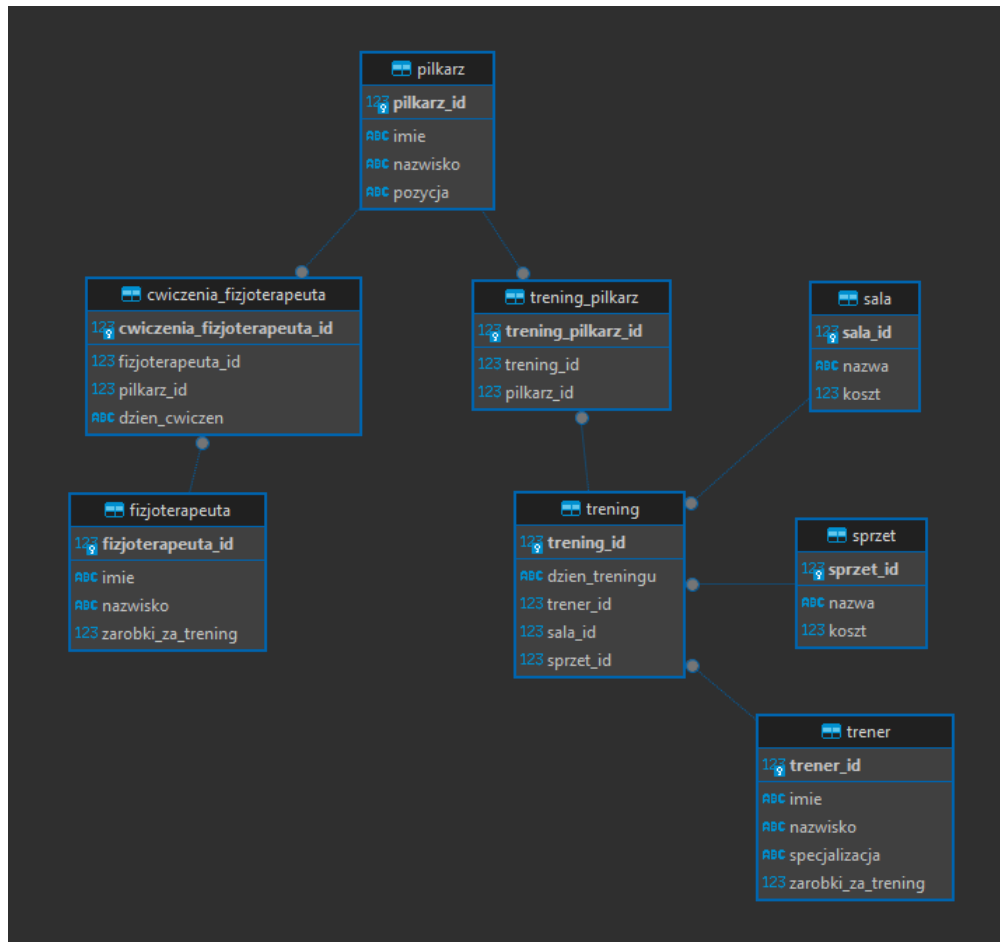
3 Tabele

- pilkarz - tabela reprezentująca piłkarza w klubie
- trener - tabela reprezentująca trenera w klubie
- sala - tabela reprezentująca sale
- sprzet - tabela reprezentująca sprzet
- trening - tabela reprezentująca trening w klubie
- fizjoterapeuta - tabela reprezentująca fizjoterapeutę w klubie
- cwiczenia_fizjoterapeuta - tabela asocjacyjna łącząca fizjoterapeutę z piłkarzem
- trening_pilkarz - tabela asocjacyjna łącząca trening z piłkarzem

4 Przepływ danych

Operacje wyświetlania są wykonane za pomocą odpowiednich zapytań wysyłanych do bazy danych. Operacje usuwania i dodawania pobierają dane z formularzy i wysyłają zapytania do bazy danych, następnie odpowiednie rekordy w tabelach są dodawane bądź usuwane.

5 Relacje między tabelami



Rysunek 1: Diagram ERD bazy danych

- fizjoterapeuta n : m pilkarz, tabele pilkarz i fizjoterapeuta są połączone w sposób n:m za pomocą tablicy asocjacyjnej cwiczenia_fizjoterapeuta. Każdy pilkarz może mieć wiele ćwiczeń tak samo jak każdy fizjoterapeuta

- sala n : 1 trening, tabele sala i trening są powiązane relacją n:1, każda sala może być przypisana do wielu treningów, natomiast każdy trening może mieć tylko jedną salę
- sprzęt n : 1 trening, tabele sprzęt i trening są powiązane relacją n:1, każdy sprzęt może być przypisany do wielu treningów, natomiast każdy trening może mieć tylko jeden sprzęt
- trener n : 1 trening, tabele trener i trening są powiązane relacją n:1, każdy trener może być przypisany do wielu treningów, natomiast każdy trening może mieć tylko jednego trenera
- trening n : m piłkarz, tabele trening i piłkarz są powiązane relacją n : m za pomocą tablicy asocjacyjnej trening_piłkarz. Każdy piłkarz może brać udział w wielu treningach tak samo jak każdy trening może mieć wielu piłkarzy

6 Atrybuty tabel

6.1 piłkarz

- piłkarz_id SERIAL PRIMARY KEY - klucz główny
- imie varchar(30) not null - imie piłkarza
- nazwisko varchar(30) not null - nazwisko piłkarza
- pozycja varchar(30) not null - pozycja piłkarza, na której gra

6.2 trener

- trener_id SERIAL PRIMARY KEY - klucz główny
- imie varchar(30) not null - imie trenera
- nazwisko varchar(30) not null - nazwisko trenera
- specjalizacja varchar(30) not null - pozycja piłkarzy, których trenuje trener
- zarobki_za_trening integer not null - zarobki trenera za jeden trening

6.3 fizjoterapeuta

- fizjoterapeuta_id SERIAL PRIMARY KEY - klucz główny
- imie varchar(30) not null - imie fizjoterapeuty
- nazwisko varchar(30) not null - nazwisko fizjoterapeuty
- zarobki_za_trening integer not null - zarobki fizjoterapeuty za pojedynczą sesję treningu fizjoterapeutycznego

6.4 sala

- sala_id SERIAL PRIMARY KEY - klucz główny
- nazwa varchar(30) NOT NULL UNIQUE - nazwa sali
- koszt integer NOT NULL - koszt wynajęcia sali

6.5 sprzęt

- sprzęt_id SERIAL PRIMARY KEY - klucz główny
- nazwa varchar(30) not null UNIQUE - nazwa sprzętu
- koszt integer not null - koszt wynajęcia sprzętu

6.6 trening

- trening_id SERIAL PRIMARY KEY - główny
- dzien_treningu varchar(30) - dzień tygodnia, w którym odbywa się trening
- trener_id integer not null - klucz obcy odnoszący się do tabeli trener
- sala_id integer not null - klucz obcy odnoszący się do tabeli sala
- sprzęt_id integer not null - klucz obcy odnoszący się do tabeli sprzęt
- FOREIGN KEY (trener_id) REFERENCES trener(trener_id)
- FOREIGN KEY (sala_id) REFERENCES sala(sala_id)
- FOREIGN KEY (sprzet_id) REFERENCES sprzęt(sprzet_id)

6.7 cwiczenia_fizjoterapeuta

- cwiczenia_fizjoterapeuta_id SERIAL PRIMARY KEY - klucz główny
- fizjoterapeuta_id integer not null - klucz obcy odnoszący się do tabeli fizjoterapeuta
- pilkarz_id integer not null - klucz obcy odnoszący się do tabeli pilkarz
- dzien_cwiczen varchar(30) not null - dzień tygodnia, w którym odbywają się ćwiczenia fizjoterapeutyczne
- FOREIGN KEY (fizjoterapeuta_id) REFERENCES fizjoterapeuta(fizjoterapeuta_id)
- FOREIGN KEY (pilkarz_id) REFERENCES pilkarz(pilkarz_id)

6.8 pilkarz_trening

- trening_pilkarz_id SERIAL PRIMARY KEY - klucz główny
- trening_id INTEGER NOT NULL - klucz obcy odnoszący się do tabeli trening
- pilkarz_id INTEGER NOT NULL - klucz obcy odnoszący się do tabeli pilkarz
- FOREIGN KEY (trening_id) REFERENCES trening(trening_id)
- FOREIGN KEY (pilkarz_id) REFERENCES pilkarz(pilkarz_id)

7 Analiza zależności funkcyjnych i normalizacja tabel

Tabele są zoptymalizowane do 3NF.

8 Triggery

8.1 Sprawdzanie pozycji

Podczas dodawania elementu do tabeli trening_pilkarz wywołuję trigger, który wywołuje funkcję sprawdz_pozycje(). Funkcja ta pobiera, za pomocą wyrażenia SELECT, pozycję na jakiej gra piłkarz oraz pozycję, którą trenuje trener biorący udział w treningu. Następnie porównuje te wartości i jeżeli są różne, wyrzuci exception.

```

CREATE FUNCTION sprawdz_pozycje()
RETURNS TRIGGER AS $$
DECLARE
    pilkarzPozycja varchar(30);
    trenerPozycja varchar(30);
BEGIN
    SELECT p.pozycja INTO pilkarzPozycja FROM pilkarz p
    join trening_pilkarz tp on p.pilkarz_id = tp.pilkarz_id
    WHERE tp.pilkarz_id = NEW.pilkarz_id;

    SELECT t.specjalizacja INTO trenerPozycja FROM trener t
    join trening tr on t.trener_id = tr.trener_id
    join trening_pilkarz tp on tr.trening_id = tp.trening_id
    WHERE tp.trening_id = NEW.trening_id;

    IF pilkarzPozycja <> trenerPozycja THEN
        RAISE EXCEPTION 'Błędne dopasowanie pozycji';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

create trigger sprawdz_pozycje_trigger
BEFORE INSERT OR UPDATE ON trening_pilkarz
FOR EACH ROW
EXECUTE PROCEDURE sprawdz_pozycje();

```


8.2 Usuwanie zależnych

Podczas usuwania rekordów z tabel pilkarz, trener, fizjoterapeuta, sala, sprzęt, trening wywoływane są triggerzy usuwające zależne od nich rekordy

8.2.1 Usuwanie piłkarza

```
CREATE FUNCTION usun_zalezne_pilkarz()
RETURNS TRIGGER AS $$
BEGIN
    DELETE FROM trening_pilkarz WHERE pilkarz_id = OLD.pilkarz_id;
    DELETE FROM cwiczenia_fizjoterapeuta WHERE pilkarz_id = OLD.pilkarz_id;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER usun_pilkarza_trigger
BEFORE DELETE ON pilkarz
FOR EACH ROW
EXECUTE FUNCTION usun_zalezne_pilkarz();
```

8.2.2 Usuwanie trenera

```
CREATE FUNCTION usun_zalezne_trener()
RETURNS TRIGGER AS $$
BEGIN
    DELETE FROM trening WHERE trener_id = OLD.trener_id;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

create trigger usun_trenera_trigger
BEFORE DELETE ON trener
FOR EACH ROW
EXECUTE FUNCTION usun_zalezne_trener();
```

8.2.3 Usuwanie fizjoterapeuta

```
CREATE FUNCTION usun_zalezne_fizjoterapeuta()  
RETURNS TRIGGER AS $$  
BEGIN  
    DELETE FROM cwiczenia_fizjoterapeuta  
    WHERE fizjoterapeuta_id = OLD.fizjoterapeuta_id;  
    RETURN OLD;  
END;  
$$ LANGUAGE plpgsql;  
  
create trigger usun_fizjoterapeute_trigger  
BEFORE DELETE ON fizjoterapeuta  
FOR EACH ROW  
EXECUTE FUNCTION usun_zalezne_fizjoterapeuta();
```

8.2.4 Usuwanie treningu

```
CREATE FUNCTION usun_zalezne_trening()  
RETURNS TRIGGER AS $$  
BEGIN  
    DELETE FROM trening_pilkarz WHERE trening_id = OLD.trening_id;  
    RETURN OLD;  
END;  
$$ LANGUAGE plpgsql;  
  
create trigger usun_trening_trigger  
BEFORE DELETE ON trening  
FOR EACH ROW  
EXECUTE FUNCTION usun_zalezne_trening();
```

8.2.5 Usuwanie sali

```
CREATE FUNCTION usun_zalezne_sala()
RETURNS TRIGGER AS $$
BEGIN
    DELETE FROM trening WHERE sala_id = OLD.sala_id;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

create trigger usun_sala_trigger
BEFORE DELETE ON sala
FOR EACH ROW
EXECUTE FUNCTION usun_zalezne_sala();
```

8.2.6 Usuwanie sprzętu

```
CREATE FUNCTION usun_zalezne_sprzet()
RETURNS TRIGGER AS $$
BEGIN
    DELETE FROM trening WHERE sprzet_id = OLD.sprzet_id;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

create trigger usun_sprzet_trigger
BEFORE DELETE ON sprzet
FOR EACH ROW
EXECUTE FUNCTION usun_zalezne_sprzet();
```

9 Widoki

9.1 Widok treningu

Widok treningu wyświetla podstawowe informacje o odbywanych treningach. Umożliwia on uzyskanie informacji o sali, sprzęcie czy trenerze prowadzącym trening

```

CREATE VIEW widok_treningu AS
SELECT
    t.trening_id,
    t.dzien_treningu as dzien_treningu,
    t.trener_id,
    tr.imie AS trener_imie,
    tr.nazwisko AS trener_nazwisko,
    tr.imie || ' ' || tr.nazwisko AS trener_imie_nazwisko,
    tr.specjalizacja,
    tr.zarobki_za_trening as zarobki_za_trening,
    s.nazwa AS nazwa_sali,
    s.koszt as koszt_sali,
    sp.nazwa AS nazwa_sprzetu,
    sp.koszt as koszt_sprzetu
FROM trening t
JOIN trener tr ON t.trener_id = tr.trener_id
JOIN sala s ON t.sala_id = s.sala_id
JOIN sprzet sp ON t.sprzet_id = sp.sprzet_id
group by t.trening_id, tr.imie, tr.nazwisko, s.nazwa, sp.nazwa,
t.dzien_treningu, tr.specjalizacja, tr.zarobki_za_trening, s.koszt,
sp.koszt;

```

9.2 Widok ćwiczeń fizjoterapeutycznych

Widok ćwiczeń fizjoterapeutycznych pozwala uzyskać informacje o prowadzonych ćwiczeniach fizjoterapeutycznych takie jak dane piłkarza i dane o fizjoterapeucie

```

CREATE VIEW widok_cwiczenia_fizjoterapeuta AS
SELECT
    cf.cwiczenia_fizjoterapeuta_id,
    cf.fizjoterapeuta_id,
    cf.pilkarz_id,
    cf.dzien_cwiczen as dzien_cwiczen,
    f.imie AS fizjoterapeuta_imie,
    f.imie || ' ' || f.nazwisko AS fizjoterapeuta_imie_nazwisko,
    f.zarobki_za_trening as zarobki_za_trening,
    p.imie AS pilkarz_imie,
    p.nazwisko AS pilkarz_nazwisko,
    p.imie || ' ' || p.nazwisko AS pilkarz_imie_nazwisko
FROM cwiczenia_fizjoterapeuta cf
JOIN fizjoterapeuta f ON cf.fizjoterapeuta_id = f.fizjoterapeuta_id
JOIN pilkarz p ON cf.pilkarz_id = p.pilkarz_id;

```

9.3 Tygodniowe wydatki na treningi

Widok wydatków tygodniowych na treningi pozwala uzyskać zsumowany tygodniowy koszt wydatków na trenerów, sale i sprzęt.

```
create view widok_wydatki_tygodniowe_treningi as
Select
SUM(tr.zarobki_za_trening+s.koszt+sp.koszt) as koszt_tygodniowy_treningow
from trening t
join trener tr on t.trener_id = tr.trener_id
join sala s on t.sala_id = s.sala_id
join sprzet sp on t.sprzet_id = sp.sprzet_id;
```

9.4 Tygodniowe wydatki na fizjoterapeutów

Widok tygodniowych wydatków na zajęcia fizjoterapeutyczne pozwala uzyskać tygodniowe zarobki poszczególnych fizjoterapeutów

```
create view widok_wydatki_tygodniowe_na_fizjoterapie as
Select SUM(f.zarobki_za_trening) as koszt_tygodniowy_cwiczen
from cwiczenia_fizjoterapeuta cf
join fizjoterapeuta f on cf.fizjoterapeuta_id = f.fizjoterapeuta_id;
```

9.5 Tygodniowe wydatki na trenerów

Widok tygodniowych wydatków na trenerów pozwala uzyskać tygodniowe zarobki trenerów, liczbę odbytych treningów oraz połączone imię z nazwiskiem.

```
create view widok_wydatki_tygodniowe_trenerzy as
SELECT SUM(tr.zarobki_za_trening)as koszt_tygodniowy,
count(tr.trener_id) as liczba_treningow,
tr.imie || ' ' || tr.nazwisko AS trener_imie_nazwisko
from trener tr
join trening t on tr.trener_id = t.trener_id
group by tr.imie, tr.nazwisko
order by koszt_tygodniowy desc;
```

9.6 Tygodniowe wydatki na sale

Widok tygodniowych wydatków na sale pozwala uzyskać tygodniowe wydatki na poszczególne sale, liczbę wynajęć oraz ich nazwę

```
CREATE VIEW widok_wydatki_tygodniowe_sala as
SELECT SUM(s.koszt)as koszt_tygodniowy,
count(s.sala_id) as liczba_wynajec,
s.nazwa as nazwa_sali
from sala s
join trening t on s.sala_id = t.sala_id
group by s.nazwa
order by koszt_tygodniowy desc;
```

9.7 Tygodniowe wydatki na sprzęt

Widok tygodniowych wydatków na sprzęt pozwala uzyskać tygodniowe wydatki na poszczególny wypożyczony sprzęt, liczbę wypożyczeń oraz jego nazwę

```
CREATE VIEW widok_wydatki_tygodniowe_sprzet as
SELECT SUM(sp.koszt)as koszt_tygodniowy,
count(sp.sprzet_id) as liczba_wypozycczen,
sp.nazwa as nazwa_sprzetu
from sprzet sp
join trening t on sp.sprzet_id = t.sprzet_id
group by sp.nazwa
order by koszt_tygodniowy desc;
```

9.8 Tygodniowe wydatki na fizjoterapeutów

Widok tygodniowych wydatków na fizjoterapeutów pozwala uzyskać tygodniowe zarobki poszczególnych fizjoterapeutów, liczbę prowadzonych ćwiczeń oraz złączone imię i nazwisko fizjoterapeuty

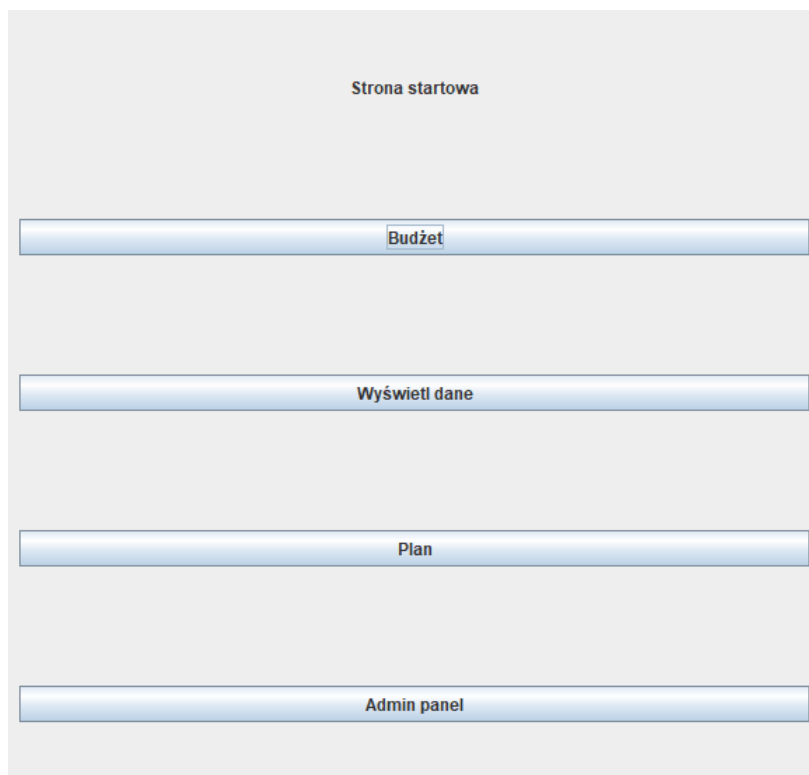
```
CREATE VIEW widok_wydatki_tygodniowe_fizjoterapeuta as
SELECT SUM(f.zarobki_za_trening)as koszt_tygodniowy,
count(f.fizjoterapeuta_id) as liczba_zajec_fizjoterapeutycznych,
f.imie || ' ' || f.nazwisko AS fizjoterapeuta_imie_nazwisko
from fizjoterapeuta f
join cwiczenia_fizjoterapeuta cf on f.fizjoterapeuta_id = cf.fizjoterapeuta_id
group by f.imie, f.nazwisko
order by koszt_tygodniowy desc;
```

10 Aplikacja

Aplikacja została wykonana w języku JAVA z wykorzystaniem biblioteki SWING. Do łączenia się z bazą danych wykorzystana została biblioteka JDBC.

10.1 Strona startowa

Na stronie startowej znajdują się 4 przyciski. Przycisk budżet prowadzi do okna, z którego możemy przeczytać szczegóły dotyczące budżetu. Wyświetl dane przenosi do okna, w którym możemy przeglądać zawartości wszystkich tabel. Plan umożliwia zobaczenie planów tygodnia konkretnej osoby oraz plan wszystkich treningów. Admin panel umożliwia dodawanie oraz usuwanie rekordów z tabel.



Rysunek 2: Strona startowa

10.2 Budżet

Po wybraniu z listy rodzaju wydatków na panelu z prawej strony wyświetlą się szczegóły tygodniowych wydatków, wraz z łączną tygodniową kwotą.

The screenshot shows a window titled "Wydatki tygodniowe". Inside, there is a table with the following content:

Tabela	
Tygodniowe wydatki na treningi	
Tygodniowe wydatki na sprzęt:	1080
Tygodniowe wydatki na sale:	18100
Tygodniowe wydatki na pensje trenerów:	11600
=====	
Łączne wydatki na treningi:	30780

Below the table, there is a dropdown menu currently set to "Treningi". The dropdown list shows the following options:

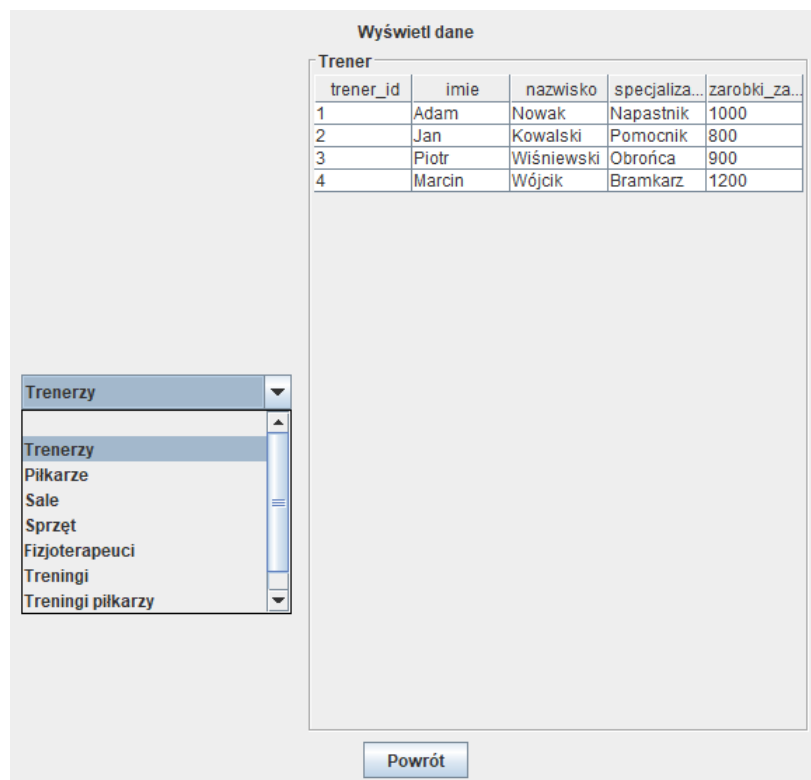
- Treningi
- Ćwiczenia
- Trenerzy
- Fizjoterapeuci
- Sale
- Sprzęt

At the bottom right of the window is a button labeled "Powrót".

Rysunek 3: Budżet

10.3 Wyświetl dane

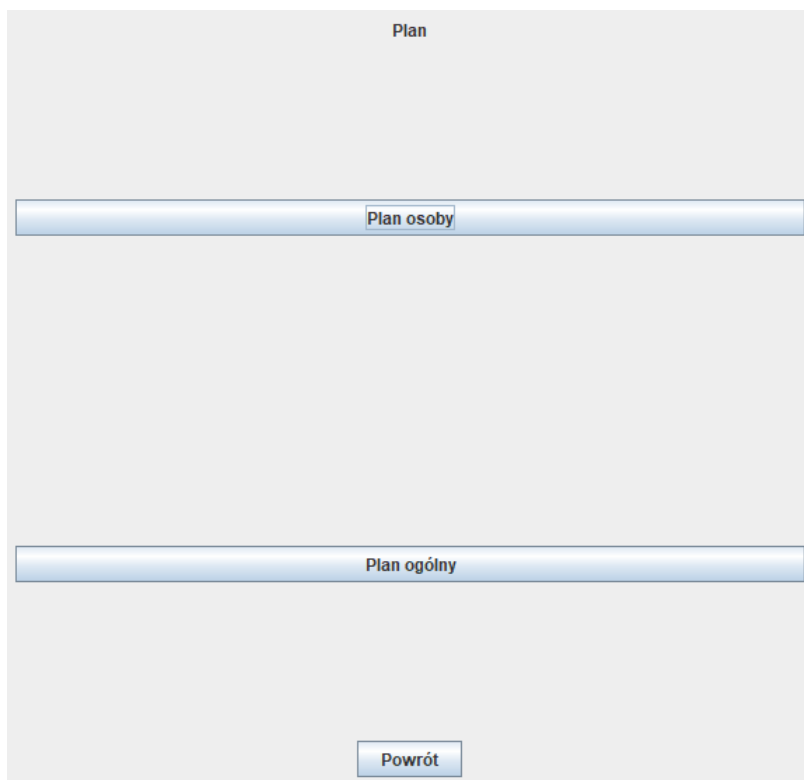
Po wybraniu z listy tabeli na panelu po prawej stronie wyświetli się cała tabela.



Rysunek 4: Wyświetl dane

10.4 Plan

Można wybrać rodzaj planu jaki nas interesuje. Plan osoby umożliwia przeglądanie planu dla konkretnej osoby, natomiast plan ogólny pokazuje plan tygodniowy zawierający wszystkie odbywające się treningi i ćwiczenia fizjoterapeutyczne.



Rysunek 5: Plan

10.5 Plan osoby

Po wybraniu rodzaju osoby, na panelu z prawej strony wyświetlą się dostępne osoby, dla których możemy wyświetlić plan. Po wpisaniu ID i przyciśnięciu przycisku 'Wyświetl' wyświetli się plan tygodnia dla wybranej osoby

The screenshot shows a web application titled "Plan Osoby". It features a grid of seven boxes representing the days of the week: **Poniedziałek** (Monday), **Wtorek** (Tuesday), **Środa** (Wednesday), **Czwartek** (Thursday), **Piątek** (Friday), **Sobota** (Saturday), and **Niedziela** (Sunday). Each box contains the text "Zajęcia z:" followed by a name and a dashed line for additional details. The names are: Monday (Jakub Moder), Wednesday (Łukasz Piszczek), Friday (Karol Linetty), and Saturday (Bartosz Bereszyński). To the right of the grid is a control panel with three radio buttons: **Fizjoterapeuta** (selected), **Trener**, and **Piłkarz**. Below these is an **ID** input field containing the number "3" and a **Wyświetl** button. Further right is a **Lista** panel titled "Fizjoterapeuci:" with a list of four names: 1 - Adam Kowalski, 2 - Michał Kaczmarek, 3 - Kamil Kowalczyk, and 4 - Grzegorz Lis. At the bottom center of the interface is a **Powrót** button.

Rysunek 6: Plan osoby

10.6 Plan ogólny

Po zaznaczeniu Treningów bądź ćwiczeń wyświetlą się one w planie tygodnia.

Plan ogólny

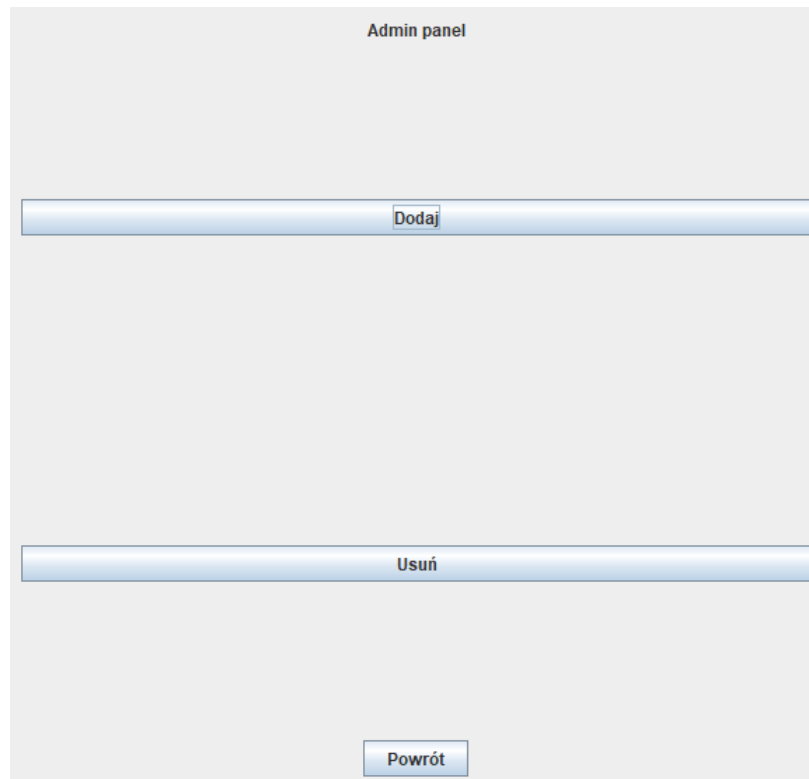
Poniedziałek Trening Prowadzący: Jan Kowalski Specjalizacja: Pomocnik Sala: Sala B Sprzęt: Hantle ----- Trening Prowadzący: Adam Nowak Specjalizacja: Napastnik Sala: Sala A Sprzęt: Piłki -----	Wtorek Trening Prowadzący: Jan Kowalski Specjalizacja: Pomocnik Sala: Sala E Sprzęt: Piłki ----- Trening Prowadzący: Marcin Wójcik Specjalizacja: Bramkarz Sala: Sala B Sprzęt: Hantle -----	Środa Trening Prowadzący: Piotr Wiśniewski Specjalizacja: Obrońca Sala: Sala C Sprzęt: Małe bramki ----- Trening Prowadzący: Marcin Wójcik Specjalizacja: Bramkarz Sala: Sala E Sprzęt: Maty do ćwiczeń -----	Czwartek Trening Prowadzący: Adam Nowak Specjalizacja: Napastnik Sala: Sala D Sprzęt: Małe bramki ----- Trening Prowadzący: Jan Kowalski Specjalizacja: Pomocnik Sala: Sala A Sprzęt: Piłki -----	<input checked="" type="checkbox"/> Treningi <input type="checkbox"/> Ćwiczenia
Piątek Trening Prowadzący: Adam Nowak Specjalizacja: Napastnik Sala: Sala B Sprzęt: Maty do ćwiczeń -----	Sobota Trening Prowadzący: Marcin Wójcik Specjalizacja: Bramkarz Sala: Sala C Sprzęt: Maty do ćwiczeń ----- Trening Prowadzący: Jan Kowalski Specjalizacja: Pomocnik Sala: Sala A Sprzęt: Hantle -----	Niedziela Trening Prowadzący: Piotr Wiśniewski Specjalizacja: Obrońca Sala: Sala E Sprzęt: Piłki -----		

Powrót

Rysunek 7: Plan ogólny

10.7 Admin

Można wybrać czy chcemy dodać rekord do tabeli czy usunąć.



The image shows a web interface titled "Admin panel" in a light blue header. Below the header, there are two large, light blue rectangular buttons. The top button is labeled "Dodaj" (Add) and the bottom button is labeled "Usuń" (Delete). At the bottom of the panel, there is a smaller, light blue button labeled "Powrót" (Return).

Rysunek 8: Admin panel

10.8 Dodaj

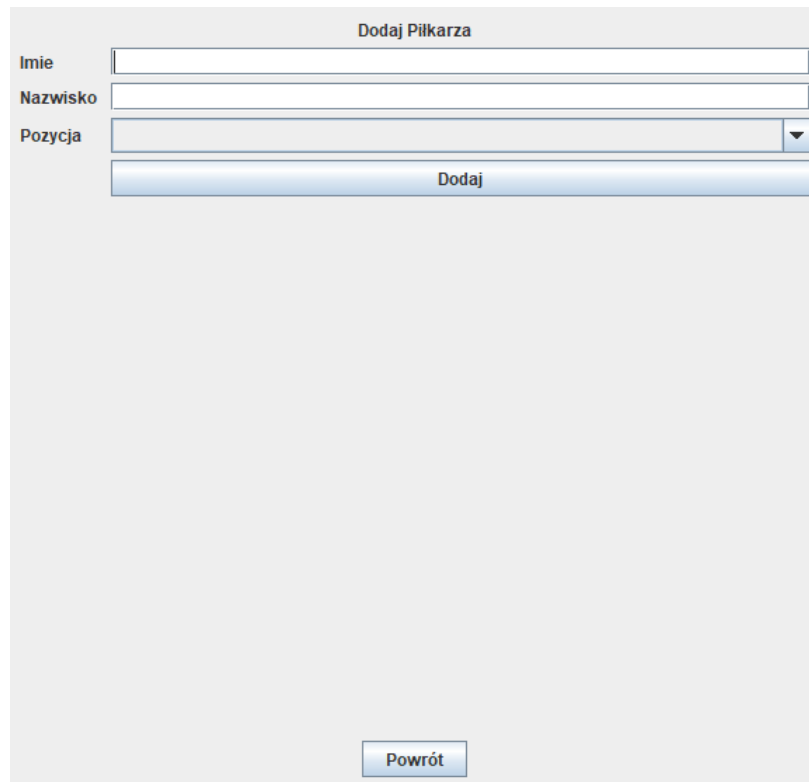
Możliwość wyboru tabeli, do której będzie dodawany rekord.

The screenshot shows a web application interface with a light gray background. At the top, centered, is the title "Dodaj". Below it is a vertical list of nine blue buttons with white text, each representing a different table or record type. The buttons are: "Piłkarz", "Trener", "Fizjoterapeuta", "Sala", "Sprzęt", "Trening", "Przypisz piłkarza do treningu", and "Ćwiczenia fizjoterapeutyczne". At the bottom center of the interface is a small blue button with white text labeled "Powrót".

Rysunek 9: Dodaj

10.9 Dodaj piłkarza

Po wpisaniu imienia i nazwiska piłkarza oraz wybraniu przycisku, po kliknięciu przycisku dodaj, rekord zostanie dodany do tabeli.



Forma dodawania piłkarza. W górnej części znajduje się tytuł "Dodaj Piłkarza". Poniżej znajdują się trzy pola tekstowe: "Imie", "Nazwisko" i "Pozycja". Pole "Pozycja" ma przycisk rozwijania. Poniżej pól znajduje się przycisk "Dodaj". W dolnej części formularza znajduje się przycisk "Powrót".

Dodaj Piłkarza	
Imie	<input type="text"/>
Nazwisko	<input type="text"/>
Pozycja	<input type="text"/> ▼
<input type="button" value="Dodaj"/>	
<input type="button" value="Powrót"/>	

Rysunek 10: Dodaj piłkarza

10.10 Dodaj trening

Po wyborze dnia tygodnia, w którym ma odbyć się trening wyświetlą się dostępni tego dnia trenerzy, sale oraz sprzęt. Następnie należy wpisać Id poszczególnych rekordów i kliknąć przycisk 'Dodaj'.

Dodaj trening

Poniedziałek ▼

TrenerID

SalaID

SprzętID

Dostępni trenerzy

3 - Piotr Wiśniewski
4 - Marcin Wójcik

Dostępne sale

3 - Sala C, koszt: 2000
4 - Sala D, koszt: 1200
5 - Sala E, koszt: 1800

Dostępny sprzęt

3 - Maty do ćwiczeń, koszt: 60
4 - Małe bramki, koszt: 130
5 - Grzybki, koszt: 20

Dodaj

Powrót

Rysunek 11: Dodaj trening

10.11 Przypisz piłkarza do treningu

Na panelach z prawej strony wyświetlają się dostępni piłkarze i treningi. Po wpisaniu Id piłkarza oraz treningu i kliknięciu przycisku dodaj piłkarz zostanie przypisany do wybranego treningu.

Dodaj piłkarza do treningu

Piłkarz ID

Trening ID

Dodaj

Piłkarze

- 1 - Robert Lewandowski, Napastnik
- 2 - Arkadiusz Milik, Napastnik
- 3 - Jakub Moder, Pomocnik
- 4 - Karol Linetty, Pomocnik
- 5 - Bartosz Bereszyński, Obrońca
- 6 - Łukasz Piszczek, Obrońca
- 7 - Wojciech Szczęsny, Bramkarz
- 8 - Sebastian Walukiewicz, Bramkarz
- 9 - elo elo, Bramkarz

Treningi

- 9 - Adam Nowak, dzień: Piątek
Specjalizacja: Napastnik
- 10 - Jan Kowalski, dzień: Sobota
Specjalizacja: Pomocnik
- 11 - Marcin Wójcik, dzień: Sobota
Specjalizacja: Bramkarz
- 12 - Piotr Wiśniewski, dzień: Niedziela
Specjalizacja: Obrońca

Powrót

Rysunek 12: Przypisz piłkarza do treningu

10.12 Dodaj ćwiczenia fizjoterapeutyczne

Na panelu z prawej strony wyświetlają się dostępni piłkarze. Po wybraniu dnia ćwiczeń wyświetlają się dostępni fizjoterapeuci, którzy nie mają ćwiczeń tego dnia.

Dodaj ćwiczenia fizjoterapeutyczne

Dzień ćwiczeń: **Poniedziałek**

Piłkarz ID:

Fizjoterapeuta ID:

Dodaj

Piłkarze

- 1 - Robert Lewandowski
- 2 - Arkadiusz Milik
- 3 - Jakub Moder
- 4 - Karol Linetty
- 5 - Bartosz Bereszyński
- 6 - Łukasz Piszczek
- 7 - Wojciech Szczęsny
- 8 - Sebastian Walukiewicz
- 9 - elo elo

Dostępni fizjoterapeuci

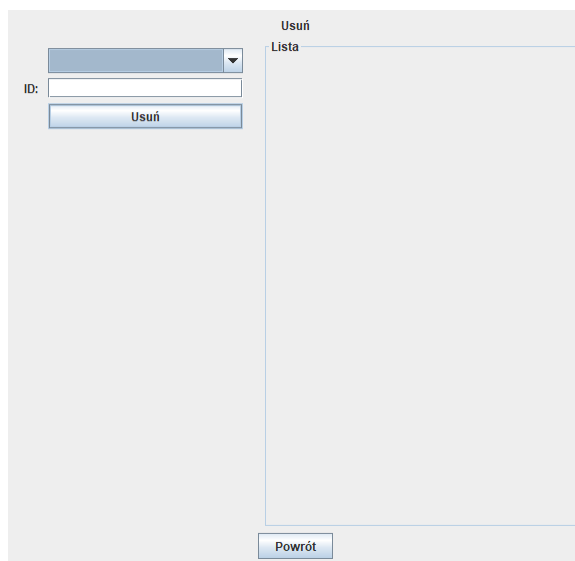
- 2 - Michał Kaczmarek
- 4 - Grzegorz Lis

Powrót

Rysunek 13: Dodaj ćwiczenia

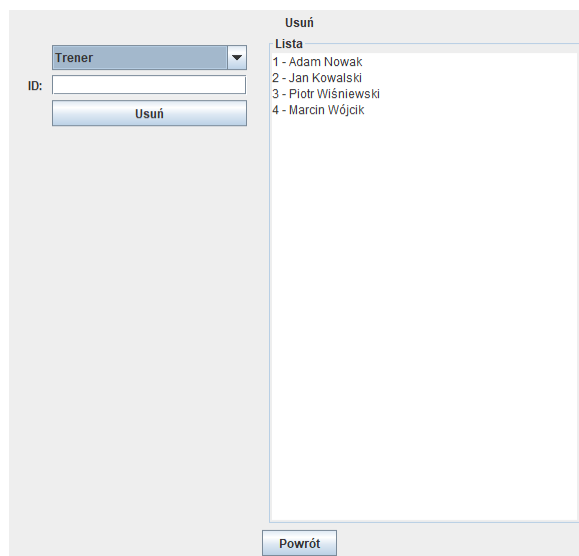
10.13 Usuń

Po wybraniu tabeli, z której chcemy usunąć rekord, na panelu z prawej strony wyświetli się lista rekordów obecnych w tabeli. Po wpisaniu Id rekordu i kliknięciu przycisku 'Usuń' rekord o wybranym Id zostanie usunięty z wybranej tabeli.



The screenshot shows a web form titled "Usuń". On the left, there is a dropdown menu, an "ID:" label, an input field, and a blue "Usuń" button. On the right, there is a large rectangular area labeled "Lista". At the bottom center, there is a blue "Powrót" button.

Rysunek 14: Usuń



The screenshot shows the same "Usuń" form as in Rysunek 14, but the "Lista" area now contains a list of four items:

- 1 - Adam Nowak
- 2 - Jan Kowalski
- 3 - Piotr Wiśniewski
- 4 - Marcin Wójcik

The other elements of the form (dropdown, ID input, "Usuń" button, and "Powrót" button) remain the same.

Rysunek 15: Usuń

11 Makropolecenia

W celu ułatwienia dodawania, przeglądania oraz usuwania danych, program wykorzystuje takie elementy jak ComboBox, które pozwalają nam wybrać dostępne opcje. Poza tym podczas dodawania oraz usuwania rekordów wyświetlane są rekordy z tabel, co znacznie ułatwia usuwanie oraz dodawanie rekordów.

12 Wprowadzanie danych

Wszystkie tabele są wypełnione danymi umożliwiającymi posługiwanie się programem. Wstawianie danych znajduje się w pliku inserty.sql. Wszystkie rekordy, poza kluczami głównymi, zostały wprowadzone ręcznie. Klucze główne są wprowadzane automatycznie za pomocą SERIAL.

13 Baza danych

Baza wykorzystuje PostgreSQL oraz umieszczona jest na hostingu ElephantSQL. Wszystkie dane potrzebne do korzystania z programu zostały już dodane.