

Dokumentacja projektu wykonywanego w ramach zajęć **BAZY DANYCH I**

I. Projekt koncepcji i założenia

1. Temat projektu.

Tematem projektu było stworzenie relacyjnej bazy danych oraz aplikacji do jej zarządzania dla dziecięcej szkoły piłkarskiej. Głównym zadaniem projektu było przydzielanie dzieci do odpowiednich grup, archiwizacja niezbędnych danych oraz poprawne zarządzanie terminami.

2. Analiza wymagań użytkownika.

Użytkownik może dodać dziecko tylko do odpowiedniej grupy. Do każdej grupy przyporządkowane są terminy treningów, trenerzy a także sala. Użytkownik ma wgląd do harmonogramów zajęć, terminów meczy planowanych, a także do statystyk meczy rozegranych oraz ilości sponsorów przyporządkowanych do każdej z grup.

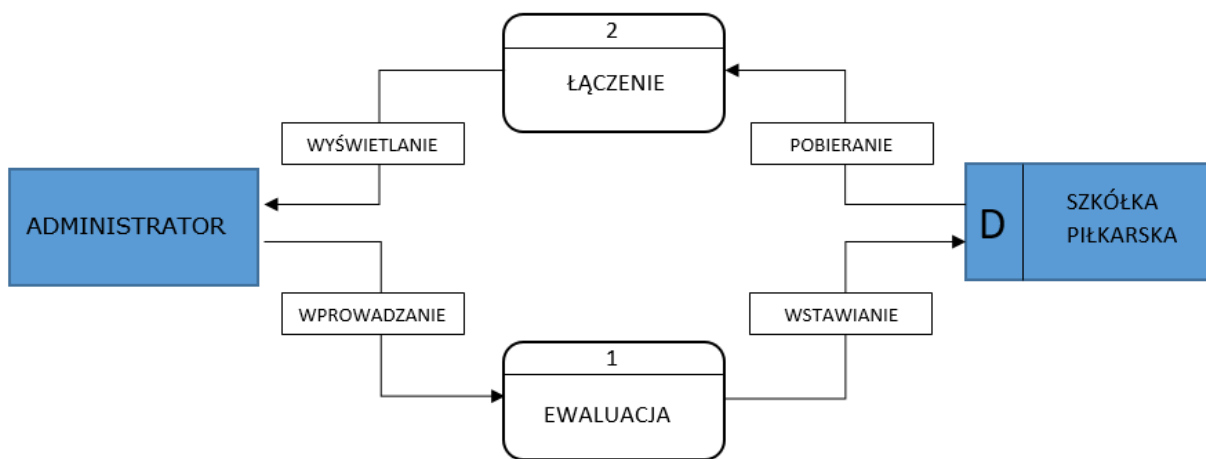
3. Zaprojektowanie funkcji.

- Wyświetlanie statystyk osiągnięć drużyn w celu analitycznych.
- Wgląd w harmonogramy drużyn, trenerów oraz grafik użytkownika sal, w celu rozsądnej organizacji zajęć.
- Kontrola dodawania dzieci do odpowiedniej grupy.
- Wgląd w ilość godzin pracy trenerów w celu sprawiedliwego rozkładu pracy.
- Wyświetlenie danych sponsorów oraz statystyk ilości sponsorów dla każdej z drużyn.
- Wgląd w pojemność drużyn w celu rozsądnego zarządzania szkołą oraz optymalizacją tworzenia nowych drużyn.

II. Projekt diagramów

1. Diagram przepływu danych (DFD).

Projekt składa się z bardzo prostej struktury przepływu danych. Dwoma głównymi składowymi są baza danych oraz interfejs służący do prostej komunikacji zapisu, odczytu oraz usuwania danych. Poprawność danych wejściowych jest kontrolowana przez bazę, jeśli żadna z założonych reguł nie jest naruszona, dane wstawiane są do odpowiednich tabel. W razie wykrycia błędu wyświetlane jest okno dialogowe z opisem problemu. Odczyt danych polega na wyświetleniu w aplikacji wszystkich tabel, oraz stworzonych w bazie raportów w formie widoków.



2. Zdefiniowanie encji (obiektów) oraz ich atrybutów.

W bazie danych powstały poniższe tabele, wszystkie pola są NOT NULL:

- Dzieci - tabela przechowująca informacje o należących do szkoły dzieciach.
 - id_dziecka - klucz główny
 - -imie i nazwisko - pola typu VARCHAR zaczynające się dużą literą
 - wiek - pole typu INTEGER, wartość z przedziału 3-18
 - data_przyjęcia - pole typu DATE w następującym formacie rok-miesiąc-dzień
 - id_pozycji - klucz obcy odwołujący się do rekordu w tabeli Pozycje
 - id_drużyny - klucz obcy odwołujący się do rekordu w tabeli Drużyny
 - id_opiekuna - klucz obcy odwołujący się do rekordu w tabeli Opiekunowie
- Opiekunowie - tabela przechowująca informacje o opiekunach dzieci należących do szkoły
 - id_opiekuna - klucz główny
 - imie i nazwisko - pola typu VARCHAR zaczynające się dużą literą
 - nr_telefonu - pole typu VARCHAR składające się z dokładnie 9 cyfr
 - nr_dowodu - pole typu VARCHAR składające się z dokładnie 3 dużych liter oraz 6 cyfr
 - adres - pole typu VARCHAR
- Pozycje - tabela słownikowa przechowująca nazwy pozycji
 - id_pozycji - klucz główny
 - nazwa_pozycji - pole typu VARCHAR
- Drużyny - tabela przechowująca informacje o drużynach
 - id_drużyny - klucz główny
 - nazwa - pole typu VARCHAR
 - id_trenera - klucz obcy odwołujący się do rekordu w tabeli Trenerzy
 - min_wiek - pole typu INTEGER, wartość z przedziału 3-18
 - max_wiek - pole typu INTEGER, wartość z przedziału 3-18, min_wiek ≤ max_wiek

- Trenerzy - tabela przechowująca informacje o trenerach
 - id_trenera - klucz główny
 - imie i nazwisko - pola typu VARCHAR zaczynające się dużą literą
 - nr_telefonu - pole typu VARCHAR składające się z dokładnie 9 cyfr
 - numer_dowodu - pole typu VARCHAR składające się z dokładnie 3 dużych liter oraz 6 cyfr
 - adres - pole typu VARCHAR

- Treningi - tabela przechowująca informacje o treningach
 - id_treningu - klucz główny
 - dzien – pole typu VARCHAR, przechowuje nazwę dnia tygodnia w którym odbywa się trening, w niedzielę szkoła jest zamknięta
 - godzina – pole typu TIME w postaci: hh:mm:ss, wymagane podawanie pełnych godzin lub półgwodkowych (np. 13:00:00, 13:30:00) ,godziny treningów w danej sali nie mogą się pokrywać, oraz godziny pracy jednego trenera nie mogą się pokrywać
 - czas_trwania – pole typu REAL, definiuje długość trwania treningu, wymagana jedna z wartości ze zbioru [1.0, 1.5, 2.0, 2.5, 3.0]
 - numer_sali – pole typu INTEGER, w szkole dostępne są dwie sale o numerach 1 oraz 2
 - id_trenera - klucz obcy odwołujący się do rekordu w tabeli Trenerzy, trener jest przypisany do danej drużyny, podanie id trenera innego niż przedzielony do danej drużyny, spowoduje że dane nie zostaną dodane
 - id_drużyny - klucz obcy odwołujący się do rekordu w tabeli Drużyny

- MeczePlanowane - tabela przechowująca informacje o meczach planowanych
 - id_m_planowanego - klucz główny
 - data - pole typu DATE, przechowuje datę przyszłą, podanie daty przeszłej będzie skutkowało nie dodaniem wiersza
 - miejsce - pole typu VARCHAR, informacja o tym, gdzie ma się odbyć mecz
 - nazwa_przeciwnika – pole typu VARCHAR
 - id_drużyny - klucz obcy odwołujący się do rekordu w tabeli Drużyny informuje, która z drużyn ma brać udział w meczu

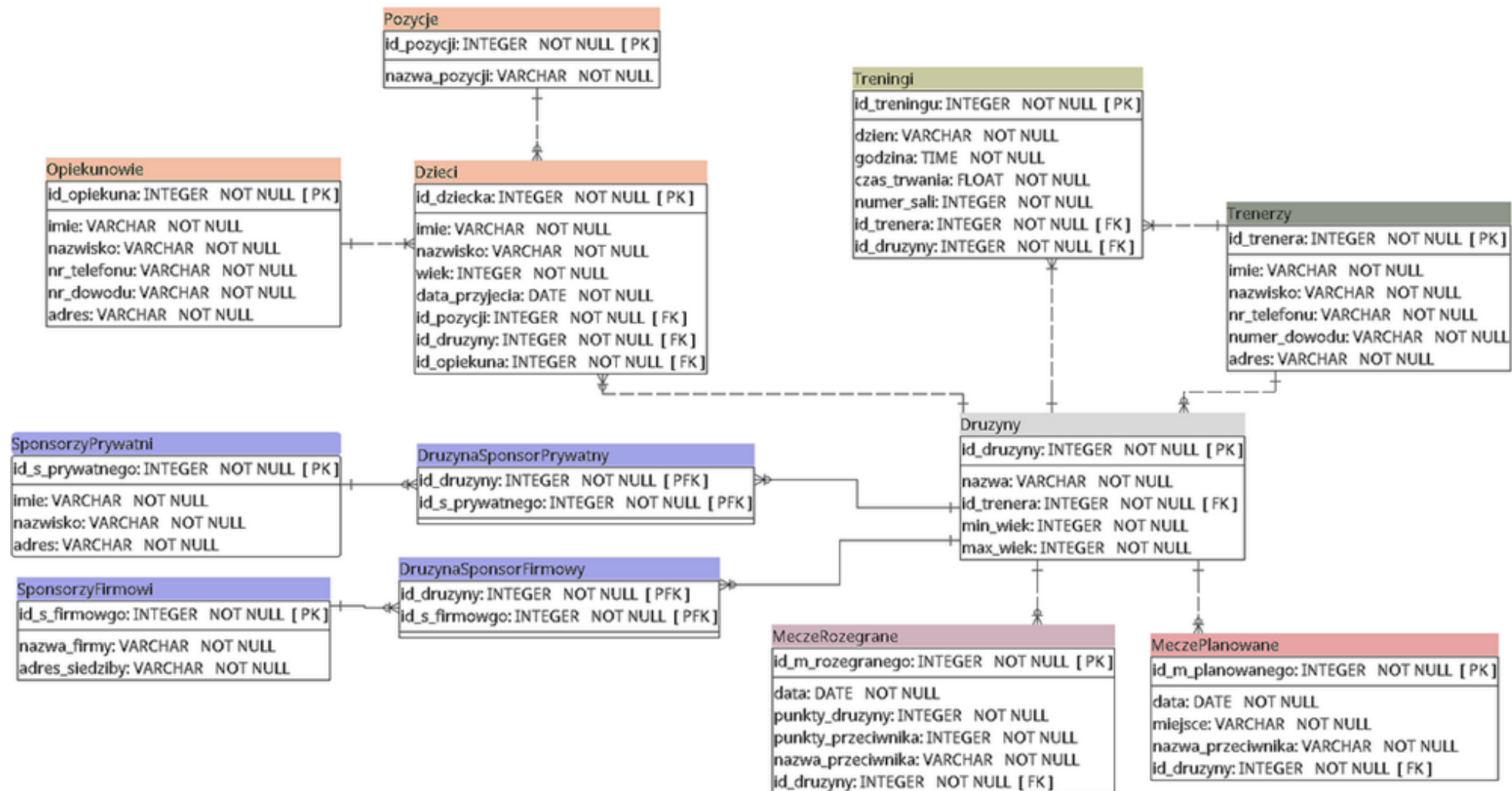
- MeczeRozegrane - tabela przechowująca informacje o meczach rozegranych
 - id_m_rozegranego - klucz główny
 - data - pole typu DATE, przechowuje datę przeszłą, podanie daty przyszłej będzie skutkowało nie dodaniem wiersza
 - punkty_drużyny – pole typu INTEGER, przechowuje ilość punktów zdobytych w meczu przez drużynę
 - punkty_przeciwnika – pole typu INTEGER, przechowuje ilość punktów zdobytych w meczu przez przeciwnika
 - nazwa_przeciwnika – pole typu VARCHAR
 - id_drużyny - klucz obcy odwołujący się do rekordu w tabeli Drużyny informuje, która z drużyn ma brać udział w meczu
 -

- SponsorzyPrywatni - tabela przechowująca informacje o sponsorach prywatnych
 - id_s_prywatnego - klucz główny
 - imie, nazwisko - pola typu VARCHAR zaczynające się dużą literą
 - adres - pole typu VARCHAR
- DruzynaSponsorPrywatny – tabela łącząca tabele Drużyny oraz SponsorzyPrywatni
 - id_drużyny – klucz obcy będący częścią klucza głównego
 - id_s_prywatnego – klucz obcy będący częścią klucza głównego
- SponsorzyFirmowi - tabela przechowująca informacje o sponsorach firmowych
 - id_s_firmowego - klucz główny
 - nazwa_firmy, adres_siedziby - pola typu VARCHAR
- DruzynaSponsorFirmowy – tabela łącząca tabele Drużyny oraz SponsorzyFirmowi
 - id_drużyny – klucz obcy będący częścią klucza głównego
 - id_s_firmowego – klucz obcy będący częścią klucza głównego

3. Zdefiniowanie relacji pomiędzy encjami.

- Dzieci – Opiekunowie, tabele powiązane relacją wiele do jeden, każde dziecko musi mieć przypisanego opiekuna, a jeden opiekun może mieć wiele dzieci
- Dzieci – Pozycje, relacja wiele do jeden – dziecko może grać na dokładnie jednej pozycji, ale na jednej pozycji może grać wiele dzieci
- Dzieci – Drużyny, relacja wiele do jeden – do jednej drużyny może należeć wiele dzieci, ale jedno dziecko może należeć do jednej drużyny
- Drużyna – SponsorzyPrywatni oraz Drużyna - SponsorzyFirmowi, tabele powiązane relacją wiele do wielu, jeden sponsor może wspierać wiele grup, oraz jedna drużyna może mieć wielu sponsorów.
- Drużyna – Trening, relacja jeden do wielu – drużyna może mieć wiele treningów, jednak do danego treningu może być przypisana tylko jedna drużyna.
- Drużyna – MeczePlanowane oraz Drużyna – MeczeRozegrane, relacje jeden do wielu – drużyna może brać udział w wielu meczach, ale mecze są rozgrywane tylko z przeciwnikami spoza szkoły.

Relacje N-M zostały wyeliminowane za pomocą tabeli łączonych.



III. Projekt logiczny.

1. Tabele, klucze, indeksy.

Powstała baza ma strukturę złożoną, relacyjną.

Na początku stworzony został schemat projektu, w którym znajduje się cała baza danych. Kolejno utworzone zostały domeny sprawdzające poprawność wprowadzanych danych. Na początku w tym schemacie za pomocą kodu wygenerowanego z programu architekt stworzone zostały tabele. Kolejno stworzone zostały wyzwalacze, które kontrolują czy wprowadzane dane są poprawne. Do tabel zostały dodane przykładowe dane. Na końcu zostały utworzone raporty w formie widoków.

2. Słowniki danych.

W projekcie znajduje się jedna tabel słownikowa przechowująca nazwy pozycji na boisku. Do tej tabeli użytkownik nie dodaje danych przez formularze ponieważ pozycje na boisku są stałe. W założeniach projektu jest ich 5. Dziedziną jest zbiór: {bramkarz, obrońca, pomocnik, skrzydłowy, napastnik}.

3. Zaprojektowanie operacji na danych.

(Załącznik plik o nazwie funkcje.sql – w pliku znajdują się komentarze do funkcji)

W projekcie powstały następujące widoki reprezentujące wyniki kwerend:

- Statystyka meczy – celem stworzenia widoku było podsumowanie osiągnięć każdej z drużyn. Zostały wypisane: ilość meczy rozegranych, ilość meczy wygranych, ilość meczy zremisowanych, ilość meczy przegranych oraz procent meczy wygranych.
- Pojemność drużyn – widok przedstawia ilość dzieci w każdej drużynie, pomocny w sprawdzeniu, czy w drużynie jest jeszcze wolne miejsce aby dodać nowe dziecko.
- Harmonogram drużyn – widok przedstawia plan zajęć dla każdej z drużyn, dzień tygodnia, początek zajęć, koniec zajęć oraz numer sali. Pomocny w szybkim sprawdzeniu kiedy odbywają się zajęcia.
- Harmonogram trenerów - widok przedstawia rozkład godzin pracy dla każdego trenera, dzień tygodnia, początek zajęć, koniec zajęć oraz numer sali. Pomocny w szybkim sprawdzeniu kiedy odbywają się zajęcia oraz podczas dodawania treningu pozwala sprawdzić czy dodawana godzina nie pokrywa się z już istniejącymi zajęciami

- Harmonogram sal – widok przedstawia rozkład godzin, w których sale są zajęte. Pomocne podczas ustalania grafiku. W szybki sposób można sprawdzić w jakich godzinach sala jest zajęta.
- Obciążenie trenerów – widok przedstawia jak dużo godzin tygodniowo ma każdy z trenerów oraz ile grup prowadzi.
- Drużyny ilość sponsorów – widok przedstawia zestawienie ilości sponsorów prywatnych, sponsorów firmowych oraz sumy sponsorów dla każdej z drużyn. Dzięki temu szybko można sprawdzić, która z drużyn musi znaleźć sponsora.

IV. Projekt funkcjonalny.

1. Interfejsy do prezentacji, edycji i obsługi danych.

Aplikacja składa się z trzech głównych zakładek: dodawanie, usuwanie, tabele oraz widoki.

W zakładce dodawanie znajduje się prosty formularz służący do dodawania rekordów do wszystkich tabel. W zakładce usuwanie znajduje się formularz do usuwania danych z tabel za pomocą id. Błędnie wprowadzone zapytania są odrzucane przez bazę a w aplikacji pojawia się okno dialogowe z odpowiednim komentarzem.

2. Wizualizacja danych.

Przechodząc do zakładki tabele oraz wybierając zakładkę z nazwą tabeli zostanie wyświetlona jej zawartość. Dane z tabeli zostały pobrane poleceniem `SELECT * FROM nazwa_tabeli`. W zakładce widoki znajdują się wyniki raportów opisanych w punkcie III.3.

Zakładki zostały wykonane po to, aby w szybki sposób przechodzić między tabelami.

V. Dokumentacja.

1. Dokumentacja użytkownika. Uruchomienie oraz obsługa aplikacji.

Kod źródłowy znajduje się w przesłanym archiwum. Aplikację należy uruchomić na własnym komputerze na systemie linux za pomocą komendy: `./skrypt.sh`. Aplikacja poprzez tunelowanie łączy się z serwerem pascal oraz bazą danych.

Po uruchomieniu aplikacji widoczny jest panel do wstawiania oraz usuwania danych. Klikając w odpowiednie zakładki wyświetlają się poszczególne tabele oraz widoki.

W zakładce „dodawanie” można dodać dane do każdej z tabeli z osobna, uzupełniając wszystkie pola poprawnymi danymi. Zapytanie wykonuje się w momencie wciśnięcia przycisku „dodaj”. Jeżeli dane są niepoprawne wyświetlany jest komunikat z informacją o błędzie. Jeśli dane są poprawne polecenie zostaje poprawnie wykonane, a pola formularza zostają wyczyszczone. W ten sam sposób należy usuwać dane.

2. Dokumentacja techniczna.

Aplikacja została napisana w języku JAVA z użyciem bibliotek Swing i AWT. Podstawowa dokumentacja techniczna znajduje się w kodzie.

3. Literatura.

<https://www.postgresql.org/docs/>

<https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>