



AGH

Akademia Górniczo-Hutnicza
im. Stanisława Staszica
w Krakowie

Bazy danych I

Dokumentacja projektu

Bartosz Mikołajczyk

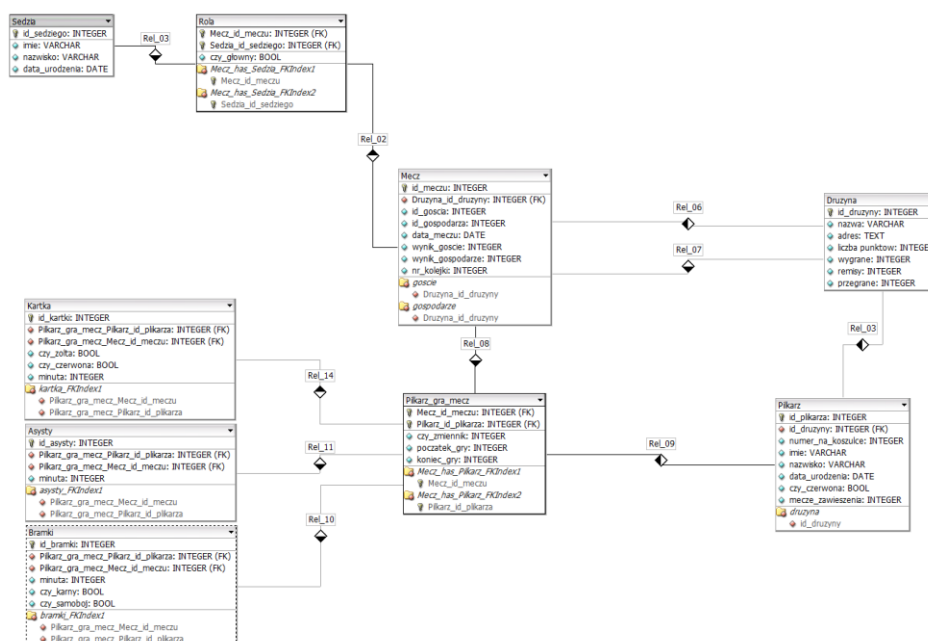
Wydział Fizyki i Informatyki Stosowanej

2020 / 2021

1. Projekt koncepcji, założenia

- Temat projektu: aplikacja zarządzająca bazą danych piłkarskiej tabeli ligowej
- Aplikacja ma na zadaniu dać klientowi możliwość utworzenia tabeli ligowej meczów w piłkę nożną. Poprzez raporty klient ma wgląd w wygląd tabeli ligowej, wypis najlepszych strzelców, aktualnie zawieszonych za kartki zawodników oraz terminarz meczów. Podstawowe funkcje zapewniane przez aplikację to wprowadzanie, usuwanie oraz edytowanie:
 - Drużyn;
 - Zawodników należących do konkretnej drużyny;
 - Meczów;
 - Sędziów;
 - Ról sędziów w poszczególnych meczach;
 - Występu zawodników w konkretnym meczu – dodanie do bazy danych bramki, asysty lub kartek takiego zawodnika.

2. Projekt diagramów (konceptualny)



Zdjęcie 1. Diagram ERD bazy danych piłkarskiej tabeli ligowej

3. Projekt logiczny

Diagram encji znajduje się na zdjęciu 1. Skrypty inicjalizujące bazę danych (plik **db.sql**), wraz z funkcjami oraz wyzwalaczami (**triggers.sql**), widokami (**views.sql**) oraz skrypt wypełniający bazę podstawowymi informacjami znajduje się w załączniku.

Encje znajdujące się w bazie danych:

- Drużyna
 - **Klucz główny**
 - Id_drużyny – zmienna typu INTEGER (SERIAL);
 - **Atrybuty**
 - Nazwa – zmienna typu VARCHAR, przechowująca nazwę drużyny, nie może być NULL;
 - Adres – zmienna typu TEXT, przechowująca adres drużyny, nie może być NULL;
 - Liczba punktów – zmienna typu INTEGER, przechowująca liczbę punktów drużyny, aktualizowana przy wprowadzaniu poszczególnych meczów, domyślnie równa 0;
 - Wygrane – zmienna typu INTEGER, przechowująca liczbę wygranych drużyny, reszta jak wyżej;
 - Remisy – zmienna typu INTEGER, przechowująca liczbę remisów drużyny, reszta jak wyżej;
 - Przegrane – zmienna typu INTEGER, przechowująca liczbę przegranych drużyny, reszta jak wyżej;
- Sędzia
 - **Klucz główny**
 - Id_sędziego – zmienna typu INTEGER (SERIAL);
 - **Atrybuty**
 - Imię – zmienna typu VARCHAR, przechowująca imię sędziego, nie może być NULL;
 - Nazwisko – zmienna typu VARCHAR, przechowująca nazwisko sędziego, nie może być NULL;

- Data urodzenia – zmienna typu DATE, przechowująca datę urodzenia sędziego, nie może być NULL;
- Piłkarz
 - **Klucz główny**
 - Id_pilkarza – zmienna typu INTEGER (SERIAL);
 - **Klucz obcy**
 - Id_druzyzny – zmienna typu INTEGER, przechowująca informację o drużynie piłkarza;
 - **Atrybuty**
 - Imie – zmienna typu VARCHAR, przechowująca imię piłkarza, nie może być NULL;
 - Nazwisko – zmienna typu VARCHAR, przechowująca nazwisko piłkarza, nie może być NULL;
 - Czy_czerwona – zmienna typu BOOLEAN, przechowująca informację o tym, czy dany piłkarz jest zawieszony, domyślnie wynosi 'False';
 - Mecze_zawieszenia – zmienna typu INTEGER, informująca ile meczów zawieszenia zostało zawodnikowi, domyślnie warta 0;
- Mecz
 - **Klucz główny**
 - Id_meczu - zmienna typu INTEGER (SERIAL);
 - **Klucz obcy**
 - Id_gospodarza – zmienna typu INTEGER zawierająca Id_druzyzny gospodarza biorącego udział w meczu;
 - Id_goscia – zmienna typu INTEGER zawierająca Id_druzyny gościa biorącego udział w meczu;
 - **Atrybuty**
 - Data_meczu – zmienna typu DATE, informująca o dacie rozegrania danego meczu;
 - Wynik_gospodarze – zmienna typu INTEGER, przechowująca informację o liczbie bramek strzelonych przez gospodarzy, domyślnie wynosi 0, aktualizowana przy wprowadzaniu kolejnych bramek;

- Wynik_goscie – zmienna typu INTEGER, przechowująca informację o liczbie bramek strzelonych przez gości, domyślnie wynosi 0, aktualizowana przy wprowadzaniu kolejnych bramek;
 - Nr_kolejki – zmienna typu INTEGER, musi być liczbą większą niż 0, informująca o numerze kolejki sezonu ligowego (jedna kolejka sezonu – jedna runda meczy rozegranych pomiędzy wszystkimi drużynami);
- Kartka
 - **Klucz główny**
 - Id_kartki – zmienna typu INTEGER (SERIAL);
 - **Klucz obcy**
 - Id_meczu – zmienna typu INTEGER;
 - Id_pilkarza – zmienna typu INTEGER;
 - Czy_zolta – zmienna typu BOOLEAN, informująca użytkownika, czy dana kartka jest żółta;
 - Czy_czerwona – zmienna typu BOOLEAN, informująca użytkownika, czy dana kartka jest czerwona;
 - Minuta – zmienna typu INTEGER, informująca w której minucie zawodnik otrzymał kartkę;
- Asysty
 - **Klucz główny**
 - Id_asysty – zmienna typu INTEGER (SERIAL);
 - **Klucz obcy**
 - Id_meczu – zmienna typu INTEGER;
 - Id_pilkarza – zmienna typu INTEGER;
 - Minuta – zmienna typu INTEGER, informująca w której minucie zawodnik zdobył asystę;
- Bramki
 - **Klucz główny**
 - Id_bramki – zmienna typu INTEGER (SERIAL);
 - **Klucz obcy**
 - Id_meczu – zmienna typu INTEGER;

- Id_pilkarza – zmienna typu INTEGER;
- Czy_karny – zmienna typu BOOLEAN, informująca użytkownika, czy dana bramka została strzelona z karnego;
- Czy_samoboj – zmienna typu BOOLEAN, informująca użytkownika, czy dana bramka jest tzw. „samobójem”, czyli golem samobójczym;
- Minuta – zmienna typu INTEGER, informująca w której minucie zawodnik zdobył bramkę
- Rola (encja asocjacyjna)
 - **Klucz główny / obcy**
 - Id_meczu – zmienna typu INTEGER;
 - Id_sedziego – zmienna typu INTEGER;
 - **Atrybuty**
 - Czy_glowny – zmienna typu BOOLEAN, informująca, czy dany sędzia jest głównym;
- Piłkarz_mecz (encja asocjacyjna)
 - **Klucz główny / obcy**
 - Id_meczu – zmienna typu INTEGER;
 - Id_pilkarza – zmienna typu INTEGER;
 - Czy_zmiennik – zmienna typu BOOLEAN informująca, czy dany zawodnik jest zmiennikiem;
 - Początek_gry – zmienna typu INTEGER, mówiąca, w której minucie zawodnik rozpoczął mecz;
 - Koniec_gry – zmienna typu INTEGER, mówiąca, w której minucie zawodnik zakończył mecz;

Widoki znajdujące się w bazie danych:

- Tabela – wypis wszystkich drużyn z pozycją w tabeli (według punktów), liczbą rozegranych meczy, wygranych, przegranych, remisów oraz liczbą piłkarzy w drużynie;
- Najlepsi strzelcy (top_scorers) – wypis wszystkich zawodników, którzy kiedykolwiek strzelili bramkę z pozycją (według liczby goli), z liczbą bramek, asyst, kartek oraz liczbą strzelonych karnych;

- Terminarz – wypis wszystkich meczy, z numerem kolejki, datą meczu, nazwami obu drużyn biorących udział w meczu, końcowym wynikiem oraz imieniem i nazwiskiem głównego sędziego;
- Zawieszenia – wypis wszystkich piłkarzy, odbywających zawieszenie z powodu otrzymania dwóch żółtych kartek lub jednej czerwonej.

Do obsługi bazy danych, zachowania spójności danych oraz zablokowania wprowadzania niepoprawnych danych wykorzystano następujące funkcje wbudowane oraz wyzwalacze:

- **Trigger_kartka** – wyzwalacz obsługujący zawieszenia w przypadku otrzymania przez zawodnika czerwonej kartki lub dwóch żółtych oraz powstrzymuje użytkownika przed wprowadzeniem niewłaściwych danych (np. trzecia kartka dla danego zawodnika, dwie czerwone kartki, zła minuta kartki);
- **Trigger_punkty1** oraz **Trigger_punkty3** – wyzwalacze obsługujące dodawanie / odejmowanie punktów zgromadzonych przez drużynę w przypadku wsadzania / usuwania danych;
- **Trigger_mecz** – wyzwalacz obsługujący zmniejszanie pozostałych meczów w zawieszeniu zawodnika przy dodawaniu kolejnych meczów;
- **Sprawdz_mecze** – wyzwalacz sprawdzający poprawność danych przy dodaniu / aktualizacji meczu, sprawdza, czy dana drużyna grała już w kolejce oraz czy dane dwie drużyny grały już ze sobą mecz (w sezonie mogą zagrać ze sobą dwa razy, zmieniając się w byciu gospodarzem – gościem);
- **Sprawdz_zawodnika** – wyzwalacz sprawdzający poprawność danych przy dodawaniu / aktualizacji danych w tabeli **piłkarz_mecz**;
- **Sprawdz_piłkarz_mecz** – wyzwalacz sprawdzający, czy przy dodawaniu / aktualizacji bramki, kartki lub asysty zostały one zamieszczone dla dobrej pozycji w tabeli **piłkarz_mecz**, w dobrej minucie (pomiędzy początek_gry a koniec_gry);
- **Sprawdz_numer** – wyzwalacz sprawdzający, czy w danej drużynie nie znajdują się dwaj zawodnicy o tym samym numerze koszulki.;
- **Sprawdz_role** – wyzwalacz sprawdzający poprawność danych dla dodania / aktualizacji danych w tabeli **rola**;
- **Sprawdz_poprawnosc_bramki** – wyzwalacz sprawdzający, czy dana bramka nie została zamieszczona zarówno jako karny i samobój;
- **Sprawdz_bramki** – wyzwalacz aktualizujący wynik danego meczu na podstawie zamieszczonej bramki oraz sprawdzający, czy w danej minucie nie padła więcej niż jedna bramka (piłkarze muszą się nacieszyć golem);

- **Zmniejsz_bramki** – wyzwalacz działający na podobnej zasadzie co powyższy, jednak zmniejszający liczbę bramek, wywoływany przy usuwaniu / edycji danych;
- **Sprawdz_asyste** – wyzwalacz sprawdzający poprawność danych dla asyst;
- **Zmien_czas** – wyzwalacz aktualizujący asysty przy usuwaniu lub edycji bramek;
- **Usun_pilkarz_mecz_dane** – wyzwalacz usuwający bramki, asysty oraz kartki dla danej pozycji w tabeli **pilkarz_mecz** przy jej usunięciu;

Do wszystkich wyzwalaczy przypisane są funkcje składowe napisane w języku *plpgsql*, dzięki którym są one w stanie egzekwować opisane wyżej funkcjonalności. Dodatkowo w bazie danych wykorzystano funkcję składową **get_main_referee**, która za argument przyjmuje id meczu, po czym zwraca imię i nazwisko głównego sędziego przypisanego do niego.

4. Projekt funkcjonalny

1

2

Football League Table

Najlepsi strzelcy

Zawieszenia

Terminarz

Tabele ▾

Pozycja	Nazwa	Rozegrano	Wygrane	Remisy	Przegrane	Punkty	Ilość piłkarzy
1	Arsenal	9	1	8	0	11	14
2	Chelsea	3	0	3	0	3	14
3	Aston Villa	2	0	2	0	2	14
4	Everton	2	0	2	0	2	14
5	Leicester City	2	0	2	0	2	14
6	Liverpool	2	0	2	0	2	14
7	Tottenham Hotspur	2	0	2	0	2	14
8	Manchester City	1	0	1	0	1	14
9	West Bromwich Albion	1	0	1	0	1	14
10	West Ham	1	0	1	0	1	14
11	Manchester United	1	0	0	1	0	14

Made by mikolajczyk
 Na przedmiot Bazy Danych 2020/2021
 Wydział Fizyki i Informatyki Stosowanej AGH

Zdjęcie 2. Strona główna aplikacji

Po otwarciu strony głównej aplikacji, użytkownikowi wyświetla się pierwszy z raportów w postaci tabeli ligowej. Pole oznaczone cyfrą 1 zawiera linki do kolejnych raportów stworzonych na podstawie wymienionych wcześniej widoków. Wszystkie raporty przedstawione są w postaci tabel. Pole 2 „Tabele” zawiera linki do dokładnych danych dla poszczególnych tabel, na których użytkownik może wykonywać operacje dodawania, usuwania lub edycji.

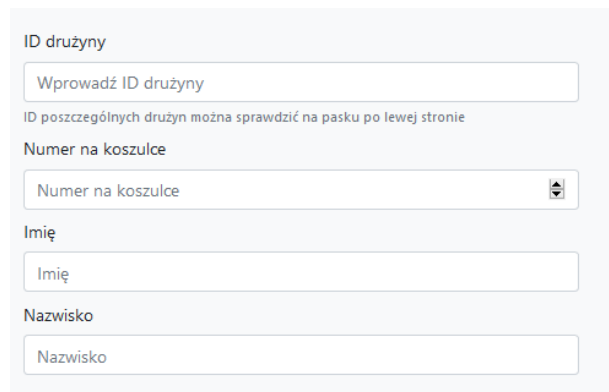
The screenshot displays the 'Football League Table' application. On the left, a sidebar titled 'Drużyny' (Teams) lists 11 teams with their IDs. The main content area is titled 'Spis piłkarzy' (Player List). At the top of the main area, there's a navigation bar with 'Football League Table', 'Najlepsi strzelcy', 'Zawieszenia', 'Terminarz', and a 'Tabele' dropdown menu. Below this, there's a button 'Wyświetl formę dodania zawodnika' (Show form to add player). The main table lists 14 players with columns for ID, team ID, jersey number, first name, and last name. Each row has 'Usuń' (Delete) and 'Edytuj' (Edit) buttons. The table is numbered 5. The sidebar is numbered 3. The 'Tabele' dropdown is numbered 2. The 'Wyświetl formę dodania zawodnika' button is numbered 4.

ID drużyny	Nazwa
1	Arsenal
2	Manchester United
3	Liverpool
4	Leicester City
5	Everton
6	Tottenham Hotspur
7	Manchester City
8	Aston Villa
9	Chelsea
10	West Ham
11	West Bromwich Albion

Id piłkarza	Id drużyny	Numer na koszulce	Imię	Nazwisko	Usuń	Edytuj
1	1	13	Bernd	Leno	Usuń	Edytuj
2	1	3	Kieran	Tierney	Usuń	Edytuj
3	1	16	Rob	Holding	Usuń	Edytuj
4	1	6	Gabriel	Magalhaes	Usuń	Edytuj
5	1	2	Hector	Bellerin	Usuń	Edytuj
6	1	18	Thomas	Partey	Usuń	Edytuj
7	1	8	Dani	Ceballos	Usuń	Edytuj
8	1	32	Emile	Smith-Rowe	Usuń	Edytuj
9	1	7	Bukayo	Saka	Usuń	Edytuj
10	1	9	Alexandre	Lacazette	Usuń	Edytuj
11	1	14	Pierre-Emerick	Aubameyang	Usuń	Edytuj
12	1	15	Ainsley	Maitland-Niles	Usuń	Edytuj
13	1	33	Mohamed	Elneny	Usuń	Edytuj
14	1	12	Willian	Borghes	Usuń	Edytuj

Zdjęcie 3. Wygląd aplikacji dla tablicy danych

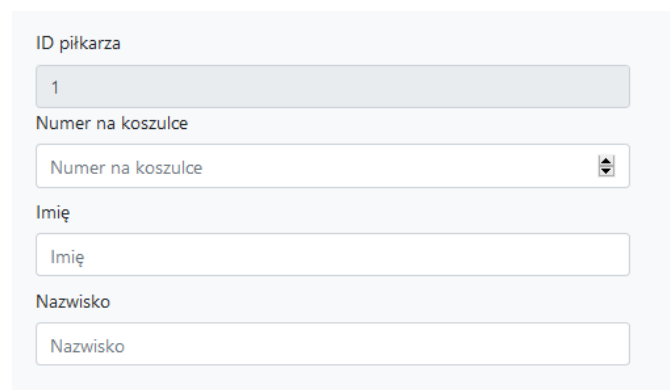
Po wejściu w link prowadzący do danej tabeli, użytkownik może dodać zawodnika (przycisk oznaczony cyfrą 4) lub przeglądać wcześniej wprowadzone dane (5, z opcją usuwania lub edycji). Do wpisywania / edycji danych w przypadku niektórych tabel umieszczono listę z danymi (3) dla ułatwienia całego procesu. Jeśli dane nie zostaną załadowane (problem z połączeniem z bazą danych), użytkownikowi pojawia się przycisk „Załaduj tabelę”, dzięki któremu może ponowić zapytanie do bazy danych (czasami trzeba spróbować parę razy, proszę o cierpliwość).



Formularz do dodawania danych dla tabeli piłkarze. Zawiera następujące pola:

- ID drużyny: Pole tekstowe z placeholderem "Wprowadź ID drużyny".
- ID poszczególnych drużyn można sprawdzić na pasku po lewej stronie: Informacja tekstowa.
- Numer na koszulce: Pole tekstowe z placeholderem "Numer na koszulce" i ikoną kursora.
- Imię: Pole tekstowe z placeholderem "Imię".
- Nazwisko: Pole tekstowe z placeholderem "Nazwisko".

Zdjęcie 4. Formularz dla dodawania danych (tabela piłkarze)



Formularz do edycji danych dla tabeli piłkarze. Zawiera następujące pola:

- ID piłkarza: Pole tekstowe z placeholderem "1".
- Numer na koszulce: Pole tekstowe z placeholderem "Numer na koszulce" i ikoną kursora.
- Imię: Pole tekstowe z placeholderem "Imię".
- Nazwisko: Pole tekstowe z placeholderem "Nazwisko".

Zdjęcie 5. Formularz do edycji danych (tabela piłkarze)

Użytkownik może wprowadzać / edytować dane przy użyciu powyższych formularzy. Po każdej operacji otrzymuje on informację zwrotną na temat poprawności danych i powodzenia operacji. Dane są wysyłane z formularzy przy użyciu metod GET, POST i DELETE. Dane tabel przechowywanych w bazie danych pobierane są dzięki zapytaniu GET, dodawanie danych jest realizowane przez metodę POST, usuwanie dzięki DELETE, a aktualizacja również dzięki POST (do url dodawane jest id pozycji, którą aktualizujemy).

5. Dokumentacja

Użytkownik może wprowadzać dane do tabel ręcznie poprzez formularze. Po uruchomieniu strony może manipulować danymi poprzez przyciski 'Usuń', 'Edytuj' oraz 'Wyświetl formę dodania'.

Aplikacja została zrealizowana przy pomocy języków HTML, CSS, JavaScript oraz NodeJS oraz framework'ów: pg (node-postgres) oraz Express. Dodatkowo strony tworzone są za pomocą języka EJS, pozwalającego generować strony HTML przy pomocy języka JavaScript.

Do stworzenia aplikacji wykorzystano następujące strony oraz literaturę:

- <https://www.w3schools.com/nodejs/>
- <https://www.postgresqltutorial.com/>
- „HTML i CSS: Podręcznik Front-End Developera”, Jon Duckett