



23.01.2020r.

**Bazy Danych**

Projekt

**Krystian Wójcik** 3EF-ZI

**158864**

1. **Określenie projektu**
2. **Określenie tematyki i zakresu projektu.**Projekt przedstawia bazę danych Klubu Jeździeckiego (Stadnina Koni, Stajnia). Zawiera on informacje dotyczące **Pracowników**, wykonujących określone zadania/zlecenia związane z Stajnią, Klientami lub Końmi, **Klientów** rezerwujących miejsce w boksach (**Rezerwacje**) lub zamawiających **Usługę** oferowaną przez Klub. Prezentowana będzie również dostępność wolnych boksów (wraz z terminem ich wynajmu/rezerwacji), **Płatności.** Do boksów w **Rezerwacji** przypisane będą konkretne **Konie**, które mogą być zarówno utrzymywane prywatnie lub przez Klub, **Klient** wynajmujący oraz przypisany mu **Pracownik**. Wszelkie informacje na temat aktualnie zleconych zadań będą przechowywane w tabeli **Usługi**, a wszystkie rezerwacje w tabeli **Rezerwacje**, które powiązane będą z konkretnym **Pracownikiem**, **Klientem**, **Koniem**, **Płatnością.** Baza składa się z 6 tabel (Płatności, Klienci, Konie, Pracownicy, Usługi, Rezerwacje). Baza umożliwia również dostęp do informacji w czasie rzeczywistym: rezerwacje **Boksów** zaczynające się i kończące w danym okresie czasu, wynajem (krótko lub długotrwała) **Koni**, dane osobowe gości Klubu (imię, nazwisko, pesel, kwota zapłacona za boks, itp.) oraz kwotę do zapłaty za daną rezerwację.
3. **Przedstawienie zagadnień związanych z tematem**Założenia naszej bazy: **Klienci** mogą korzystać z dowolnej ilości usług (nieograniczona ilość wynajmowanych boksów w **Rezerwacji,** zależna jedynie od ilości wolnych miejsc, goście “losowi”, którzy nie utrzymują **Koni** w pensjonacie, aczkolwiek korzystają z oferowanych przez Klub **Usług**. Płatność dokonywana jest na koniec każdego okresu rozliczeniowego (domyślnie płatność comiesięczna).
4. **Określenie funkcji bazy danych i ich priorytetu.**Baza danych służyć będzie zoptymalizowaniu i ułatwieniu procesu wynajmu oraz zarządzania Końmi/Rezerwacjami, a także aktualnie zarejestrowanymi Usługami. Pomoże to zoptymalizować pracę Pracowników, monitorując ich aktualne zlecenia oraz ilość przypisanych do nich klientów.
5. **Wybór technologii i typu bazy danych do zrealizowania projektu.**Projekt zostanie zrealizowany w technologii mysql, bazując na doświadczeniu zdobytym w wcześniejszych latach. Baza danych będzie bazą złożoną, relacyjną.
6. **Wybór narzędzi do zrealizowania projektu.**Projekt zostanie wykonany w online’owym narzędziu PHPMyAdmin, bazującym na strukturze MySQL. Link do narzędzia: <http://mysql.turbohost.pl/>
7. **Prezentacja przygotowanego repozytorium z opisem**

Repozytorium utworzone do realizacji projektu dostępne jest pod adresem:

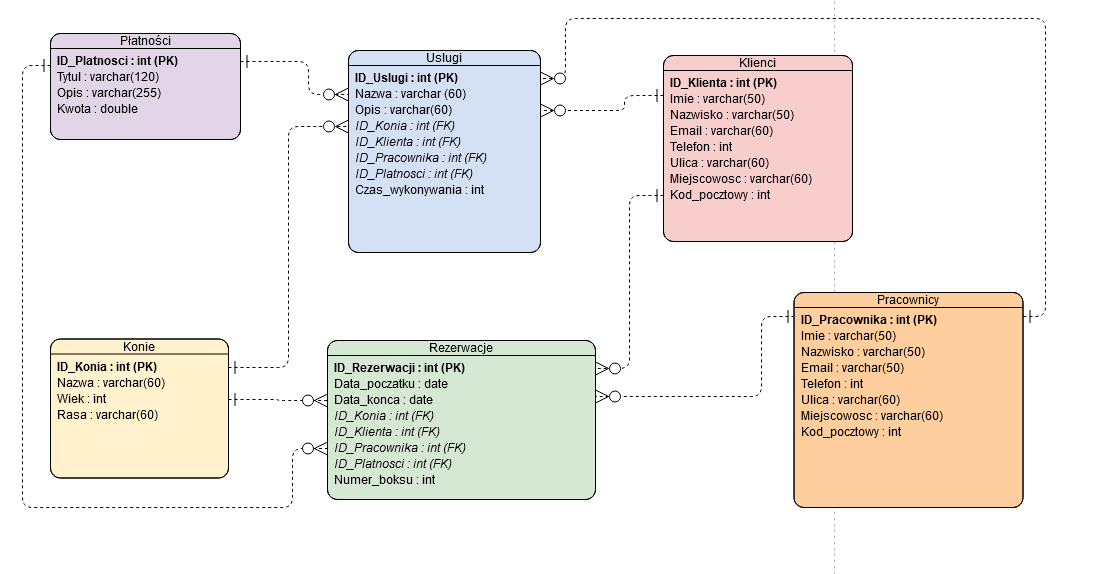
<https://github.com/WojcikKrystian/BazyDanychP>

Utworzone zostało nowe repozytorium, pozwalające przeglądać historię dodawania oraz edycji plików.

1. **Prezentacja diagramu DB**
2. **Prezentacja diagramu bazy danych.**

Diagram Bazy Danych projektu dostępny jest pod adresem:

https://github.com/WojcikKrystian/BazyDanychP/blob/master/Projekt\_IO\_Diagram.PNG



1. **Opis tabel bazy danych i ich funkcji.**

**Omówienie tabel:**

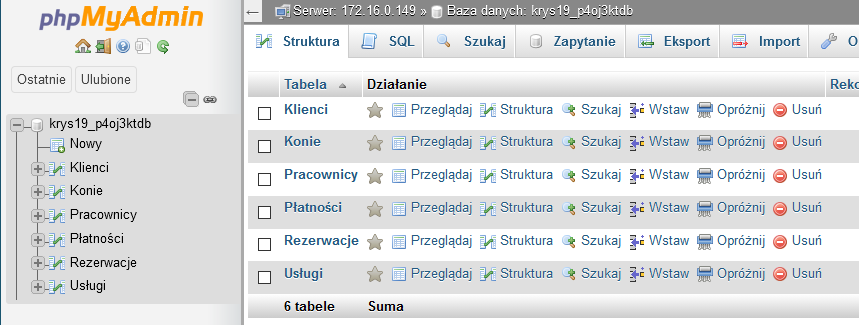
* **Klienci** – tabela zawiera dane osobowe osób dokonujących *Rezerwacji* lub wykupujących *Usługę*, numery rezerwacji oraz numery boksów/koni im przydzielonych.
* **Rezerwacje** – tabela zawiera daty początku i końca rezerwacji (które mogą być dokonywane z wyprzedzeniem), numery boksów, *Koni* na które dokonywana jest rezerwacja, numer danej rezerwacji, numer osoby rezerwującej (*Klienta*) oraz opiekuna (*Pracownika*) przypisanego do Zlecenia.
* **Płatności** – tabela zawiera informacje o kwotach jakie występują w klubie. Kwoty przypisane są do tabeli *Rezerwacje* oraz *Usługi.*
* **Konie** - tabela zawierająca podstawowe informacje o Koniu, powiązana jest z *Rezerwacjami* (w którym Boksie się znajduje, jaki Pracownik się nim opiekuje i jaki klient dokonuje rezerwacji) oraz czy aktualnie ma zarezerwowaną *Usługę*.
* **Pracownicy** - tabela zawierająca dane osobowe osób pracujących w Klubie. Powiązani są z *Usługami* oraz *Rezerwacjami*, a także pośrednio z Klientami których obsługują.
* **Usługi** – tabela zawiera informacje o usługach z jakich korzystają poszczególni Klienci, Pracownikach obsługujących zlecenie, ich cenę i czas wykonywania usługi. Klient może wybrać kilka takich samych usług dla jednego Konia, lub jedną usługę dla kilku Koni.

1. **Prezentacja opisów bazy danych.**

Baza danych została utworzona w systemie phpMyAdmin pozwalając w sposób wizualny oraz tekstowy nią zarządzać. Dodane zostały tabele zawarte w diagramie ERD wraz z przykładowymi wpisami. Tabele zostały powiązane kluczami podstawowymi oraz obcymi, wyeliminowane zostały możliwe połączenia wiele-do-wielu.

1. **Prezentacja problemów w realizacji.**

Problemem w realizacji, było nieestetyczne ustawienie Tabeli w prezentowanym diagramie, co zostało poprawione. Pierw mylny pomysł powiązania koni z dodatkową tabelą Boksy został rozwiązany poprzez połączenie tabel Boksy oraz Rezerwacje w jedną tabelę.

1. **Prezentacja SQL**
2. **Prezentacja wykonania bazy danych.**Projekt dostępny do edycji pod adresem: <http://mysql.turbohost.pl>  
   Dane logowania znajdują się w repozytorium: <https://github.com/WojcikKrystian/BazyDanychP/blob/master/BazaDanych.txt>  
     
   
3. **Prezentacja funkcji DB realizowanych przez SQL.**Dzięki bazie danych możemy w trybie rzeczywistym sprawdzać aktualnie wykonane rezerwacje poszczególnych Koni przez Klientów. Możemy również zauważyć, jakie usługi zostały już zrealizowane, a także jakie są do wykonania w ciągu najbliższych dni.  
   Pracownicy mogą sprawdzać co czeka ich w ciągu najbliższych dni, a także jakie są ich obowiązki. Możemy również monitorować aktualny oraz systematyczny zysk generowany przez Stajnie, dzięki sprawdzaniu dochodów oraz odejmowaniu kosztów.
4. **Prezentacja problemów w realizacji.**

Problemem związanym z aktualnym tematem było błędne wprowadzanie plików na system kontroli wersji (GitHub). Aktualnie nowe pliki dodawane są w Commitach, co pozwala na weryfikację ich wcześniejszej wersji. Zoptymalizowane zostały również encje w tabelach, pozwalając w łatwiejszy sposób przeglądać/uzupełniać dane (np. Numer telefonu zmieniony z int na varchar).

1. **Obrona projektu**
2. **Prezentacja bazy danych i głównych funkcjonalności (SQL)**

**Wyświetlanie Danych osobowych klientów oraz ilości posiadanych przez nich koni w Rezerwacji.**   


*Select K.Imie, K.Nazwisko, count(R.ID\_Rezerwacji) as Ilosc\_koni   
from Klienci K inner join Rezerwacje R on K.ID\_Klienta=R.ID\_Klienta   
GROUP BY R.ID\_Klienta;*

**Wyświetlanie wszystkich zarejestrowanych Usług przypisanych do danego Klienta. Zapytanie wyświetla Imię oraz nazwisko Klienta, Dane osobowe przypisanego do zlecenia pracownika, Tytuł zlecenia oraz termin wykonania usługi posortowany od najwcześniejszego.**



*Select K.Imie as Kli\_Imie, K.Nazwisko as Kli\_Nazwisko, P.Tytul, Pra.Imie as Pra\_Imie, Pra.Nazwisko as Pra\_Nazwisko, U.Data\_uslugi from Usługi U*

*Inner join Płatności P on P.ID\_Platnosci=U.ID\_Platnosci*

*Inner join Klienci K on K.ID\_Klienta=U.ID\_Klienta*

*Inner join Pracownicy Pra on Pra.ID\_Pracownika=U.ID\_Pracownika*

*Order by Data\_uslugi;*

**Wyświetlanie informacji o zakupionych przez klientów usługach oraz ich kwocie.**

**Zapytanie wyświetla Dane osobowe klienta, Tytuł wykupionej usługi oraz kwotę należną do zapłaty.**



*Select K.Imie as Kli\_Imie, K.Nazwisko as Kli\_Nazwisko, P.Tytul, P.Kwota from Usługi U*

*Inner join Płatności P on P.ID\_Platnosci=U.ID\_Platnosci*

*Inner join Klienci K on K.ID\_Klienta=U.ID\_Klienta*

*Order by K.Imie;*

**Zmodyfikowane poprzednie zapytanie, sumujące totalną kwotę którą klient ma uiścić do Klubu za wszystkie wykupione usługi. Zapytanie wyświetla Dane osobowe klientów oraz zsumowane należności, posegregowane od największej.**



*Select K.Imie as Kli\_Imie, K.Nazwisko as Kli\_Nazwisko, sum(P.Kwota) as Należności from Usługi U*

*Inner join Płatności P on P.ID\_Platnosci=U.ID\_Platnosci*

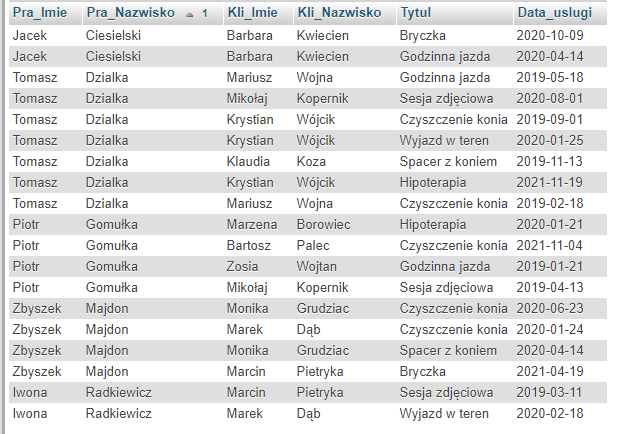
*Inner join Klienci K on K.ID\_Klienta=U.ID\_Klienta*

*Group by K.Imie*

*Order by Należności desc;*

**Wyświetlanie informacji o Kliencie, Tytule zleconej usługi oraz dacie wykonania przypisanej do konkretnego pracownika, posortowane nazwiskami.**

**Zapytanie wyświetla Dane osobowe Pracowników oraz Klientów, Tytuł realizacji oraz termin jej wykonania.**



*Select Pra.Imie as Pra\_Imie, Pra.Nazwisko as Pra\_Nazwisko, K.Imie as Kli\_Imie, K.Nazwisko as Kli\_Nazwisko, P.Tytul, U.Data\_uslugi from Usługi U*

*Inner join Płatności P on P.ID\_Platnosci=U.ID\_Platnosci*

*Inner join Klienci K on K.ID\_Klienta=U.ID\_Klienta*

*Inner join Pracownicy Pra on Pra.ID\_Pracownika=U.ID\_Pracownika*

*order by Pra\_Nazwisko;*

**Procedura pozwalająca dodawać Pracowników do bazy danych.**

*CREATE PROCEDURE `DodajPracownika`(IN `ID\_Pracownika` INT(11), IN `Imie` VARCHAR(50), IN `Nazwisko` VARCHAR(50), IN `Email` VARCHAR(60), IN `Telefon` VARCHAR(11), IN `Ulica` VARCHAR(60), IN `Miejscowosc` VARCHAR(60), IN `Kod\_pocztowy` VARCHAR(11))*

*NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER*

*BEGIN*

*insert into Pracownicy values(ID\_Pracownika, Imie, Nazwisko, Email, Telefon, Ulica, Miejscowosc, Kod\_pocztowy);*

*END;*

**Wywołanie utworzonej procedury DodajPracownika**

**Sposób 1.**

*SET @p0='6'; SET @p1='Józiu'; SET @p2='Mączka'; SET @p3='j.maczka@stajnia.pl'; SET @p4='233 434 123'; SET @p5='Poznańska 23'; SET @p6='Rzeszów'; SET @p7='35-001'; CALL `DodajPracownika`(@p0, @p1, @p2, @p3, @p4, @p5, @p6, @p7);*

**Sposób 2 (optymalny).**

*call DodajPracownika('6','Józiu’, ‘Mączka’, ‘j.maczka@stajnia.pl’, ‘233 434 123’, ‘Poznańska 23’, ‘Rzeszów’, ‘35-001’);*

**Procedura pozwalająca usuwać Pracowników z bazy danych na podstawie wybranego ID.**

*CREATE PROCEDURE `UsunPracownika`(IN `ID\_Pracownika` INT) NOT DETERMINISTIC NO SQL SQL SECURITY DEFINER BEGIN delete from Pracownicy where Pracownicy.ID\_Pracownika=ID\_Pracownika; END;*

**Wywołanie utworzonej procedury UsunPracownika**

*call UsunPracownika (6);*

**Procedura pozwalająca aktualizować tabelę Usługi wybierając ID usługi.**

*CREATE PROCEDURE `AktualizacjaUslugi`(IN `Id\_Uslugi` INT(11), IN `Data\_uslugi` DATE, IN `ID\_Konia` INT(11), IN `ID\_Klienta` INT(11), IN `ID\_Pracownika` INT(11), IN `ID\_Platnosci` INT(11))*

*NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER*

*BEGIN*

*update Usługi SET*

*Usługi.Data\_uslugi=Data\_Uslugi,*

*Usługi.ID\_Konia=ID\_Konia,*

*Usługi.ID\_Klienta=ID\_Klienta,*

*Usługi.ID\_Pracownika=ID\_Pracownika,*

*Usługi.ID\_Platnosci=ID\_Platnosci*

*where Usługi.ID\_Usługi=Id\_Uslugi; end;*

Projekt w pełni zrealizowany przez **Krystian Wójcik**Lab03 **158864**