

Piotr Ponichtera IS 3  
Dokumentacja projektu - Bazy danych I

Projekt dostępny jest na serwerze Pascal pod adresem

[http://pascal.fis.agh.edu.pl/~3ponichtera/Bazy\\_danych\\_projekt/](http://pascal.fis.agh.edu.pl/~3ponichtera/Bazy_danych_projekt/)

Aby się zalogować trzeba wpisać login: admin

hasło: admin1

Strukturę bazy można przenieść do swojej bazy danych. Plik baza.sql w katalogu SQL zawiera całą strukturę bazy danych. Po użyciu komendy `\i baza.sql` (będąc w katalogu SQL) w swojej bazie tworzą się wszystkie tabele, a także funkcje, widoki, triggerzy (z pliku zapytania.sql). Tabele są także wypełniane z pliku dane.sql.

Aby korzystać z aplikacji we własnej bazie danych należy w pliku model2.php (katalog baza/class) zmienić pola `$this->user`, `$this->pass` oraz `$this->db` i nadać im swoje dane. W moim przypadku ten fragment kodu wygląda tak.

```
$this->user="u3ponichtera";  
$this->pass="3ponichtera";  
$this->db="u3ponichtera";
```

```
self::$db = new PDO("pgsql:dbname=$this->db;user=$this->user;password=$this->pass");
```

```
self::$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

Taką samą czynność trzeba wykonać w także w pliku `dodaj_auto.tpl` (katalog template).

## 1. Projekt koncepcji, założenia

a) Naszym zadaniem było stworzenie bazy danych, która miała reprezentować konkretny problem. Tematem mojego projektu jest wypożyczalnia samochodów. Stworzona przeze mnie baza danych pozwala na kontrolowanie i zarządzanie taką wypożyczalnią. Pozwala ona na dodawanie klientów, aut, wypożyczeń, kontrolę bilansu oddziału firmy, kontrolę usterek aut i wiele innych działań. Aplikacja przeznaczona jest dla jednej osoby odpowiedzialnej za jej zarządzanie.

b) Funkcjonalności jakie miała spełniać moja baza danych miały pozwalać na wygodne korzystanie z aplikacji przez użytkownika i oferować mu wszystkie operacje potrzebne do funkcjonowania wypożyczalni samochodów. Najważniejsze operacje takie jak dodawanie klientów, aut, usuwanie rekordów oraz sam mechanizm wypożyczania zostały zaimplementowane. Oprócz tych opcji istnieje wiele innych działań ulepszających funkcjonowanie wypożyczalni.

c) Podstawową funkcją bazy jest możliwość wypożyczania samochodu. Zostało to zaimplementowane w funkcji `postgresql "zamow"`, która korzystając z innych funkcji obliczających koszt wypożyczenia, dodaje rekord do tabeli wypożyczeń oraz zmienia stan auta z dostępnego na niedostępny. Kolejną ważną funkcją jest śledzenie bilansu danego oddziału. Zostało to dodane do funkcji, w których brany pod uwagę jest koszt. Reszta funkcji daje ogólne możliwości jak na przykład wyszukiwanie konkretnego rekordu, przeglądanie historii wypożyczeń czy napraw. Do wyświetlania większości danych użyłem widoków z uwagi na to iż aby otrzymać pełną informację o na przykład aucie musiałem czerpać informacje z wielu tabel. Triggerów użyłem do prowadzenia statystyk ilości transakcji (wszystkie operacje gdzie zmieniany jest bilans oddziału).

## 2. Projekt diagramów

### a) Mój projekt zawiera 9 tabel

Tabela adres odpowiedzialna jest za przechowywanie adresów klientów oraz oddziałów. Zawiera ona pola takie jak id(klucz), miasto, województwo, numer domu, numer mieszkania.

Tabela klient odpowiedzialna jest za przechowywanie danych klienta. Zawiera ona pola takie jak id(klucz), imię, nazwisko, telefon, numer dowodu osobistego oraz klucze obce adresu oraz wypożyczenia.

Tabela oddział odpowiedzialna jest za przechowywanie danych oddziału. Zawiera ona pola takie jak id(klucz), imię I nazwisko kierownika oddziału, telefon, bilans oraz klucz obcy adresu.

Tabela auto odpowiedzialna jest za przechowywanie danych auta. Zawiera ona pola takie jak id(klucz), moc, pojemność bagażnika, cenę za dzień wypożyczenia, dostępność oraz klucze obce typu auta, marki auta, wypożyczenia

Tabela wypożyczenie odpowiedzialna jest za przechowywanie danych wypożyczenia. Zawiera ona pola takie jak id(klucz), datę wypożyczenia, datę zwrotu, koszt wypożyczenia oraz klucze obce klienta I auta.

Tabela marka auta jest tabelą słownikową reprezentującą dany model auta. Zawiera pola takie jak id(klucz), marka auta I model auta. Rekordy te są unikalne.

Tabela typ auta jest tabelą słownikową reprezentującą dany typ auta. Zawiera pola takie jak id(klucz) I typ auta.

Tabela Auto naprawa jest tabelą asocjacyjną łączącą auta z naprawą relacją n:m. Zawiera ona pola takie jak id naprawy(klucz), klucz obcy id auta, klucz obcy id rodzaju naprawy oraz datę naprawy.

Tabela Naprawa odpowiedzialna jest za przechowywanie rodzaju I kosztu naprawy. Zawiera ona pola takie jak id(klucz), rodzaj I koszt.

b) Relacja adres-klient to relacja 1:1 – każdy klient ma 1 adres I każdy adres ma jednego klienta/oddział.

Relacja adres-oddział to relacja 1:1 – każdy oddział ma 1 adres I każdy adres ma jeden oddział/klienta.

Relacja oddział auto to relacja 1:n – każdy oddział może mieć wiele aut, ale każde auto należy do jednego oddziału.

Relacja klient-wypożyczenie to relacja 1:n – każdy klient może wiele razy wypożyczać, ale konkretne wypożyczenie odnosi się do jednego klienta.

Relacja auto-marka\_auta to relacja 1:n – każde auto jest konkretnej marki, ale ta marka może produkować wiele aut.

Relacja auto-typ\_auta to relacja 1:n – każde auto jest konkretnego typu, ale ten tego typu może być wiele innych aut.

Tabela Auto\_naprawa jest tabelą asocjacyjną I pełni funkcję łącznika między autem a naprawą, aby ta relacja była n:m ponieważ każde auto może mieć wiele usterek I każda usterka może dotyczyć wielu aut.

Diagram ERD został dołączony w folderze.

Słowniki danych w mojej bazie odpowiadają za typ(np. sedan, suv) auta oraz jego model(np. Porsche Panamera). Ich ograniczeniem jest to że muszą być unikalne.

Wszystkie pliki SQL z komentarzami dołączone w folderze.

#### 4. Projekt funkcjonalny

a) Do wprowadzania danych użyłem struktury <form> języka HTML, a następnie metodą \_POST wysyłałem te dane dalej, gdzie były walidowane( walidacja danych także przeprowadzana w bazie) I przetwarzane.

b) Raporty generowane przez bazę wyświetlane są w postaci tabeli gdzie pierwszy wiersz odpowiada nazwie kolumny, a kolejne wiersze to wartości danej kolumny. Jest to wyświetlane w czytelny sposób.

c) Projekt wykonany został w technologii MVC. Po wejściu na stronę ukazuje się pole logowania, w które trzeba wpisać hasło I login. Następnie użytkownik przenoszony jest do strony głównej. Po lewej stronie jako odnośniki może wybierać poszczególne opcje, a po wyborze na środku ekranu przedstawione zostaną tabele, formularze odpowiednie dla danej opcji.

d) Projekt nie przewiduje makropoleceń. Wszystkie operacje dostępne są w kilku kliknięciach I ich działanie jest jasno opisane.

#### 5. Dokumentacja

a) Startowe dane wprowadzane są z pliku "dane.sql". Następnie z poziomu przeglądarki można dodawać dowolne dane do każdej z tabel.

b) Aby dodać wypożyczenie należy wybrać z menu po lewej stronie opcję "Wypożycz auto" następnie ukazuje się formularz w którym wybieramy ID auta z tabeli dostępnych aut, która znajduje się poniżej formularza, następnie wybieramy ID klienta z tabeli klientów, która znajduje się pod formularzem, kolejnym krokiem jest wybór daty wypożyczenia. Kiedy klikniemy na pole odpowiedzialne za wczytanie tej informacji pojawi się kalendarz, który po kliknięciu na datę dnia wprowadzi dane do pola z datą wypożyczenia w odpowiednim formacie. Założeniem projektu jest, że nie ma możliwości rezerwacji auta itp. Wypożyczamy auto tego samego dnia, w którym dodajemy ten fakt do bazy. Kolejnym polem jest data zwrotu auta, po kliknięciu także pojawi się kalendarz z którego wybieramy datę późniejszą niż data początkowa. Jeśli wybierzemy datę wcześniejszą nie przejdzie ona przez walidację w bazie danych (tak samo gdy jako datę początkową weźmiemy inną niż datę aktualnego dnia).

Wszystkie inne funkcjonalności działają w taki sam sposób. Jeżeli użytkownik ma wybierać auto to tabela tych aut wyświetla się też na stronie więc nie trzeba szukać jej gdzieś indziej.

Rekordy z tabeli słownikowej na przykład podczas dodawania nowego auta wyświetlane są za pomocą rozwijalnej listy, z której wybieramy konkretny model, czy typ. Istnieje możliwość rozszerzania tabel słownikowych.

Dokumentacja techniczna kodu PHP wykonana została aplikacją Apigen. Dostępna jest w folderze Dokumentacja\_php (dokumentacja kodu php wyświetlana w przeglądarce), a także z poziomu menu projektu (2 opcja od dołu). Dokumentacja techniczna kodu JS oraz SQL znajduje się w plikach z kodem źródłowym plików js I sql. Pliki JS znajdują się w folderze js/baza.js. Pliki PHP odpowiedzialne za funkcjonalność aplikacji znajdują się w folderze class/baza/baza.php oraz class/baza/model2.php.