

Dokumentacja projektu wykonywanego w ramach zajęć BAZY DANYCH I

Aleksandra Holik

19 I 2017

1 Projekt koncepcji, założenia

1.1 Zdefiniowanie tematu projektu

Temat projektu:

Oprogramowanie wspierające działalność firmy świadczącej przewozy pasażerskie.

Celem aplikacji jest gromadzenie i porządkowanie informacji o działalności firmy tj. zakupie usług, czyli rezerwacji biletów na przejazd lub rezerwacji pojazdów przez firmy bądź klientów. Odpowiada również za ewidencję pracowników firmy i sprzętu, co ułatwi kontrolę nad budżetem oraz zapewni najlepszą jakość świadczonych klientom usług

1.2 Analiza wymagań użytkownika

Użytkownik będzie mógł zbadać konta pracowników firmy jak również klientów posiadających konta w bazie firmy. Sprawdzi również dział sprzętowy. Zgromadzona zostanie baza danych samochodów np. kiedy dokonano zakupu danego pojazdu i jak często jest eksploatowany. Pojawi się możliwość sprawdzenia tras i ustalonych kursów firmy. Użytkownik zostanie zobowiązany do kontroli bazy rezerwacji i wypożyczeń. Będzie mógł również sprawdzić promocje proponowane przez firmę.

1.3 Zaprojektowanie funkcji

- Dodanie pojazdu, pracownika, klienta, tras
- Edycja informacji o pracownikach, pojazdach, klientach
- Wyszukiwanie ogólnych i wyspecjalizowanych informacji z poszczególnych tabel
- Wyświetlanie informacji poprzez wprowadzenie odpowiednich identyfikatorów

2 Projekt diagramów

2.1 Budowa diagramu przepływu danych

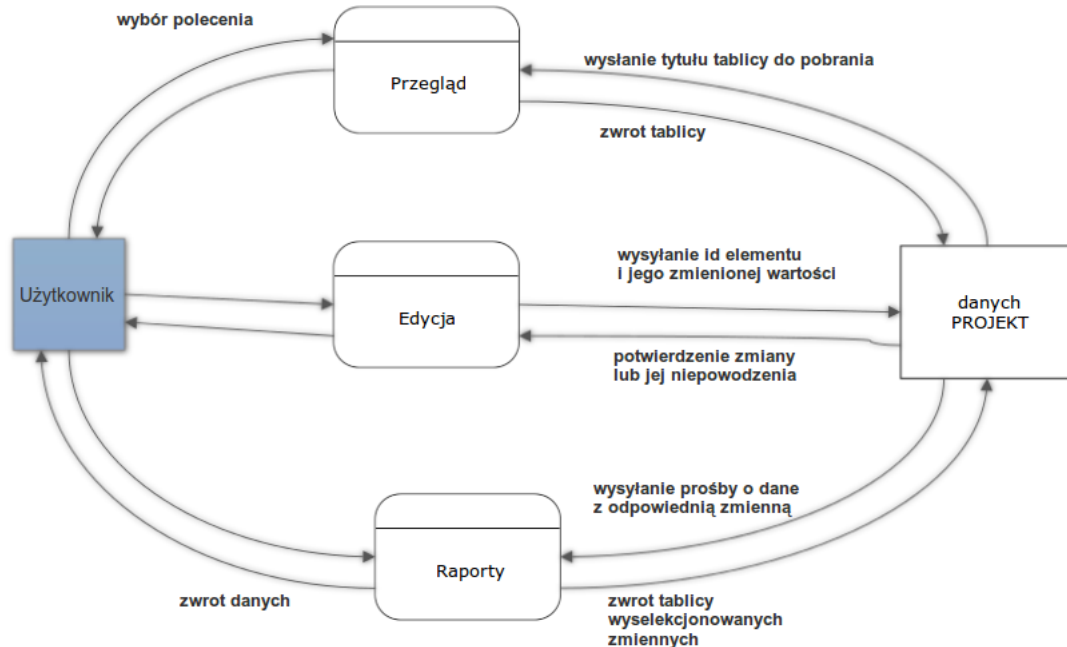


Figure 1: Diagram DFD.

2.2 Definicja encji oraz ich atrybutów

Baza danych jest zawarta w schemacie "projekt". Składa się z 8 encji

- *Kierowca* - podstawowe dane o kierowcach zatrudnionych w firmie przewozowej.
Atrybuty: imię, nazwisko, pesel, mail, data zatrudnienia;
- *Pojazd* - dane o pojeździe wykorzystywanym w firmie do przewozów klientów lub wypożyczeń.
Atrybuty: marka, data produkcji, data zakupu, liczba miejsc, data przeglądu, cena wypożyczenia, pożyczony - czy jest przeznaczony do wypożyczenia;
- *Trasa* - dane o trasach, na których kursują busy firmy.
Atrybuty: miasto początkowe, miasto końcowe, odległość, cena przejazdu;
- *Promocja* - proponowane przez firmę zniżki dla stałych klientów.
Atrybuty: nazwa, ilość darmowych przejazdów;
- *Klient* - baza klientów zapisanych do firmy. Jest powiązana z encją *Promocja*, dzięki której można się dowiedzieć jaka promocja przysługuje klientowi o konkretnym ID.
Atrybuty: imię, nazwisko, pesel, e-mail, telefon, firma, przejechane trasy;

- *Kurs* - baza zawierająca kursy ustalone na konkretne dni. Jest powiązana z encjami *Pojazd*, *Kierowca*, *Trasa*, dzięki którym może skompresować najważniejsze informacje o usłudze w jednym miejscu.
Atrybuty: liczba pasażerów rezerwujących miejsce na kurs, godzina i data wyjazdu, godzina i data wyjazdu;
- *Rezerwacja* - baza rezerwacji dokonanych przez klientów zapisanych do bazy. Jest powiązana z tablicą danych *Kurs*, *Trasa*, *Kierowca*, *Pojazd*, co nie pozwoli na błędne przypisanie do kursu.
Atrybuty: liczba zarezerwowanych miejsc;
- *Wypożyczenie* - baza zanotowanych wypożyczeń dokonanych przez klientów firmy. Powiązana jest z encjami *Klient*, *Pojazd*, *Kierowca*.
Atrybuty: data wypożyczenia;

Wszystkie encje można modyfikować w aplikacji. Należy jednak dokładnie przeczytać dokumentację o poprawnym wprowadzaniu danych, gdyż aplikacja nie zapewnia pełnej kontroli dodawanych danych. Po dodaniu lub zmianie danych warto sprawdzić rezultat działania w zakładce *Przegląd* wybierając odpowiednią tablicę.

2.3 Relacje pomiędzy encjami

Diagram relacji encji schematu *projekt*:

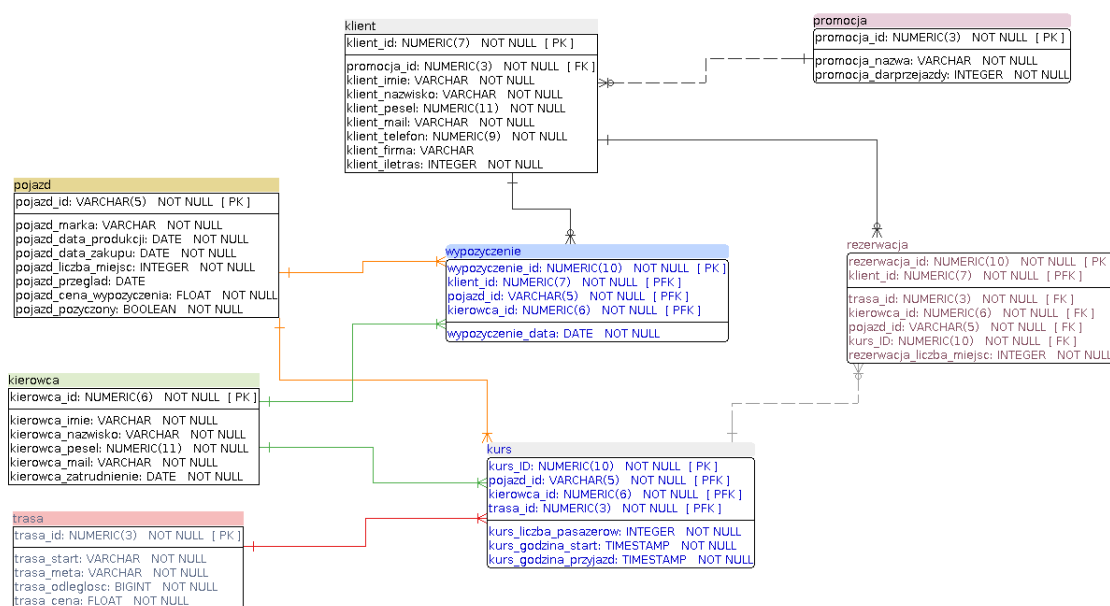


Figure 2: Diagram ERD.

3 Projekt logiczny

3.1 Zaprojektowane tabele, klucze, indeksy

Stworzono najpierw odpowiedni schemat do łatwej identyfikacji encji a następnie wszystkie encje. Na koniec na podstawie encji utworzono widok, bardzo ważny element bazy. Umożliwia on spis najważniejszych informacji o kursie: dane kierowcy, dane pojazdu, dane kursu, czas przejazdu oraz liczbę zarezerwowanych miejsc. Bazę wypełniono przykładowymi danymi, aby zaprezentować działanie aplikacji.

```
CREATE SEQUENCE projekt.klient_klient_id_seq;

CREATE TABLE projekt.Klient (
    Klient_ID NUMERIC(7) NOT NULL DEFAULT
nextval('projekt.klient_klient_id_seq'),
    Promocja_ID NUMERIC(3) NOT NULL DEFAULT 999,
    Imie VARCHAR NOT NULL,
    Nazwisko VARCHAR NOT NULL,
    Pasaż NUMERIC(11) NOT NULL,
    Mail VARCHAR NOT NULL,
    Telefon NUMERIC(9) NOT NULL,
    Firma VARCHAR,
    Ile_Tras INTEGER DEFAULT 0 NOT NULL,
    CONSTRAINT klient_id_pk PRIMARY KEY (Klient_ID)
);

ALTER SEQUENCE projekt.klient_klient_id_seq RESTART WITH 1000000
OWNED BY projekt.Klient.Klient_ID;
```

```
CREATE SEQUENCE projekt.promocja_promocja_id_seq;

CREATE TABLE projekt.Promocja (
    Promocja_ID NUMERIC(3) NOT NULL DEFAULT
nextval('projekt.promocja_promocja_id_seq'),
    Nazwa VARCHAR NOT NULL,
    Darmowe_Prażajdy INTEGER DEFAULT 0 NOT NULL,
    CONSTRAINT promocja_id_pk PRIMARY KEY (Promocja_ID)
);

ALTER SEQUENCE projekt.promocja_promocja_id_seq RESTART WITH 100
OWNED BY projekt.Promocja.Promocja_ID;
```

```
CREATE SEQUENCE projekt.rezerwacja_rezerwacja_id_seq;

CREATE TABLE projekt.Rezerwacja (
    Rezerwacja_ID NUMERIC(10) NOT NULL DEFAULT
nextval('projekt.rezerwacja_rezerwacja_id_seq'),
    Klient_ID NUMERIC(7) NOT NULL,
    Trasa_ID NUMERIC(3) NOT NULL,
    Kierowca_ID NUMERIC(6) NOT NULL,
    Pojazd_ID NUMERIC(5) NOT NULL,
    Kurs_ID NUMERIC(10) NOT NULL,
    Liczba_Miejsc INTEGER DEFAULT 1 NOT NULL,
    CONSTRAINT rezerwacja_id_pk PRIMARY KEY (Rezerwacja_ID,
Klient_ID)
);

ALTER SEQUENCE projekt.rezerwacja_rezerwacja_id_seq OWNED BY
projekt.Rezerwacja.Rezerwacja_ID;

CREATE SEQUENCE projekt.wypozyczenie_wypozyczenie_id_seq;

CREATE TABLE projekt.Wypozyczenie (
    Wypozyczenie_ID NUMERIC(10) NOT NULL DEFAULT
nextval('projekt.wypozyczenie_wypozyczenie_id_seq'),
    Klient_ID NUMERIC(7) NOT NULL,
    Pojazd_ID NUMERIC(5) NOT NULL,
    Kierowca_ID NUMERIC(6) NOT NULL,
    Data DATE NOT NULL,
    CONSTRAINT wypozyczenie_pk PRIMARY KEY (Wypozyczenie_ID,
Klient_ID, Pojazd_ID, Kierowca_ID)
);

ALTER SEQUENCE projekt.wypozyczenie_wypozyczenie_id_seq OWNED BY
projekt.Wypozyczenie.Wypozyczenie_ID;
```

```
CREATE SEQUENCE projekt.klient_klient_id_seq;

CREATE TABLE projekt.Klient (
    Klient_ID NUMERIC(7) NOT NULL DEFAULT
nextval('projekt.klient_klient_id_seq'),
    Promocja_ID NUMERIC(3) NOT NULL DEFAULT 999,
    Imie VARCHAR NOT NULL,
    Nazwisko VARCHAR NOT NULL,
    Pasaż NUMERIC(11) NOT NULL,
    Mail VARCHAR NOT NULL,
    Telefon NUMERIC(9) NOT NULL,
    Firma VARCHAR,
    Ile_Tras INTEGER DEFAULT 0 NOT NULL,
    CONSTRAINT klient_id_pk PRIMARY KEY (Klient_ID)
);

ALTER SEQUENCE projekt.klient_klient_id_seq RESTART WITH 1000000
OWNED BY projekt.Klient.Klient_ID;
```

```
CREATE SEQUENCE projekt.promocja_promocja_id_seq;

CREATE TABLE projekt.Promocja (
    Promocja_ID NUMERIC(3) NOT NULL DEFAULT
nextval('projekt.promocja_promocja_id_seq'),
    Nazwa VARCHAR NOT NULL,
    Darmowe_Prażajdy INTEGER DEFAULT 0 NOT NULL,
    CONSTRAINT promocja_id_pk PRIMARY KEY (Promocja_ID)
);

ALTER SEQUENCE projekt.promocja_promocja_id_seq RESTART WITH 100
OWNED BY projekt.Promocja.Promocja_ID;
```

```
CREATE SEQUENCE projekt.rezerwacja_rezerwacja_id_seq;

CREATE TABLE projekt.Rezerwacja (
    Rezerwacja_ID NUMERIC(10) NOT NULL DEFAULT
nextval('projekt.rezerwacja_rezerwacja_id_seq'),
    Klient_ID NUMERIC(7) NOT NULL,
    Trasa_ID NUMERIC(3) NOT NULL,
    Kierowca_ID NUMERIC(6) NOT NULL,
    Pojazd_ID NUMERIC(5) NOT NULL,
    Kurs_ID NUMERIC(10) NOT NULL,
    Liczba_Miejsc INTEGER DEFAULT 1 NOT NULL,
    CONSTRAINT rezerwacja_id_pk PRIMARY KEY (Rezerwacja_ID,
Klient_ID)
);

ALTER SEQUENCE projekt.rezerwacja_rezerwacja_id_seq OWNED BY
projekt.Rezerwacja.Rezerwacja_ID;

CREATE SEQUENCE projekt.wypozyczenie_wypozyczenie_id_seq;

CREATE TABLE projekt.Wypozyczenie (
    Wypozyczenie_ID NUMERIC(10) NOT NULL DEFAULT
nextval('projekt.wypozyczenie_wypozyczenie_id_seq'),
    Klient_ID NUMERIC(7) NOT NULL,
    Pojazd_ID NUMERIC(5) NOT NULL,
    Kierowca_ID NUMERIC(6) NOT NULL,
    Data DATE NOT NULL,
    CONSTRAINT wypozyczenie_pk PRIMARY KEY (Wypozyczenie_ID,
Klient_ID, Pojazd_ID, Kierowca_ID)
);

ALTER SEQUENCE projekt.wypozyczenie_wypozyczenie_id_seq OWNED BY
projekt.Wypozyczenie.Wypozyczenie_ID;
```

```
ALTER TABLE projekt.Klient ADD CONSTRAINT promocja_klient_fk
FOREIGN KEY (Promocja_ID)
REFERENCES projekt.Promocja (Promocja_ID)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE projekt.Kurs ADD CONSTRAINT trasa_kurs_fk
FOREIGN KEY (Trasa_ID)
REFERENCES projekt.Trasa (Trasa_ID)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE projekt.Wypozyczenie ADD CONSTRAINT
kierowca_wypozyczenie_fk
FOREIGN KEY (Kierowca_ID)
REFERENCES projekt.Kierowca (Kierowca_ID)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE projekt.Kurs ADD CONSTRAINT kierowca_kurs_fk
FOREIGN KEY (Kierowca_ID)
REFERENCES projekt.Kierowca (Kierowca_ID)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE projekt.Wypozyczenie ADD CONSTRAINT
pojazd_wypozyczenie_fk
FOREIGN KEY (Pojazd_ID)
REFERENCES projekt.Pojazd (Pojazd_ID)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE projekt.Kurs ADD CONSTRAINT pojazd_kurs_fk
FOREIGN KEY (Pojazd_ID)
REFERENCES projekt.Pojazd (Pojazd_ID)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE projekt.Rezerwacja ADD CONSTRAINT kurs_rezerwacja_fk
FOREIGN KEY (Kurs_ID, Kierowca_ID, Trasa_ID, Pojazd_ID)
REFERENCES projekt.Kurs (Kurs_ID, Kierowca_ID, Trasa_ID, Pojazd_ID)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

/* widok kurs - kierowca I */
create view projekt.KursKierowca as select kurs_id, imie, nazwisko from projekt.kurs,
projekt.kierowca where projekt.kurs.kierowca_id = projekt.kierowca.kierowca_id;

/* widok kurs - pojazd II */
create view projekt.KursPojazd as select kurs_id, marka from
projekt.kurs,projekt.pojazd where projekt.kurs.pojazd_id =
projekt.pojazd.pojazd_id;

/* widok tras - kurs */
create view projekt.KursTrasa as select kurs_id, trasa_start, godzina_wyjazdu,
trasa_meta, godzina_przyjazdu, liczba_pasazerow from projekt.kurs, projekt.trasa
where projekt.kurs.trasa_id = projekt.trasa.trasa_id;

/* łączenie pierwszych dwóch tabel */
create view projekt.KursKierowcaPojazd as select imie, nazwisko,
projekt.kurspojazd.* from projekt.KursKierowca, projekt.KursPojazd where
projekt.KursKierowca.kurs_id = projekt.KursPojazd.kurs_id;

/* całość */
create view projekt.DanyKurs as select projekt.KursTrasa.kurs_id, imie, nazwisko,
marka, trasa_start, godzina_wyjazdu, trasa_meta, godzina_przyjazdu,
liczba_pasazerow from projekt.KursKierowcaPojazd, projekt.KursTrasa where
projekt.KursTrasa.kurs_id = projekt.KursKierowcaPojazd.kurs_id
```

3.2 Słownik danych

projekt.trasa			
trasa_id	NUMERIC(3)	NOT NULL PK	DEFAULT nextval('projekt.trasa_trasa_id_seq')
trasa_start	VARCHAR	NOT NULL	
trasa_meta	VARCHAR	NOT NULL	
odleglosc	BIGINT	NOT NULL	
cena	FLOAT	NOT NULL	
projekt.kierowca			
kierowca_id	NUMERIC(6)	NOT NULL PK	DEFAULT nextval('projekt.kierowca_kierowca_id_seq')
imie	VARCHAR	NOT NULL	
nazwisko	VARCHAR	NOT NULL	
pesel	NUMERIC(11)	NOT NULL	
mail	VARCHAR	NOT NULL	
data_zatrudnienia	DATE	NOT NULL	
projekt.pojazd			
pojazd_id	NUMERIC(5)	NOT NULL PK	DEFAULT nextval('projekt.pojazd_pojazd_id_seq')
marka	VARCHAR	NOT NULL	
data_produkcji	DATE	NOT NULL	
data_zakupu	DATE	NOT NULL	
liczba_miejsc	INTEGER	NOT NULL	
przeglad	DATE		
cena_wypozyczenia	REAL	NOT NULL	
pożyczony	BOOLEAN	NOT NULL	
projekt.kurs			
kurs_id	NUMERIC(10)	NOT NULL PK	DEFAULT nextval('projekt.kurs_kurs_id_seq')
pojazd_id	NUMERIC(5)	NOT NULL	
kierowca_id	NUMERIC(6)	NOT NULL	
trasa_id	NUMERIC(3)	NOT NULL	
liczba_pasazerow	INTEGER	NOT NULL	DEFAULT 0
godzina_wyjazdu	TIMESTAMP	NOT NULL	
godzina_przyjazdu	TIMESTAMP	NOT NULL	
projekt.klient			
klient_id	NUMERIC(7)	NOT NULL	DEFAULT nextval('projekt.klient_klient_id_seq'),
promocja_id	NUMERIC(3)	NOT NULL	DEFAULT 999,
imie	VARCHAR	NOT NULL	
nazwisko	VARCHAR	NOT NULL	
pesel	NUMERIC(11)	NOT NULL	
mail	VARCHAR	NOT NULL	
telefon	NUMERIC(9)	NOT NULL	
firma	VARCHAR		
ile_tras	INTEGER	NOT NULL	DEFAULT 0
projekt.promocja			
promocja_id	NUMERIC(3)	NOT NULL	DEFAULT nextval('projekt.promocja_promocja_id_seq')
nazwa	VARCHAR	NOT NULL	
damowe_przejazdy	INTEGER	NOT NULL	DEFAULT 0
projekt.rezerwacja			
rezerwacja_id	NUMERIC(10)	NOT NULL	DEFAULT nextval('projekt.rezerwacja_rezerwacja_id_seq')
klient_id	NUMERIC(7)	NOT NULL	
trasa_id	NUMERIC(3)	NOT NULL	
kierowca_id	NUMERIC(6)	NOT NULL	
pojazd_id	NUMERIC(5)	NOT NULL	
kurs_id	NUMERIC(10)	NOT NULL	
liczba_miejsc	INTEGER	NOT NULL	DEFAULT 1
projekt.wypozyczenie			
klient_id	NUMERIC(10)	NOT NULL	DEFAULT nextval('projekt.wypozyczenie_wypozyczenie_id_seq')
pojazd_id	NUMERIC(7)	NOT NULL	
kierowca_id	NUMERIC(6)	NOT NULL	
data	DATE	NOT NULL	

3.3 Zaprojektowane operacje na danych

Raporty z bazy:

OGÓLNE DANE KURSU POBRANE Z WIDOKU projekt.danykurs

```
select * from projekt.DanyKurs where godzina_wyjazdu > '2017-01-08 00:00' and  
godzina_wyjazdu < '2017-01-09 00:00';
```

POBRANIE LISTY REZERWACJI Z DANYMI KLIENTÓW ZA PODANIEM ID KURSU

```
select distinct imie, nazwisko, liczba_miejsc from projekt.Rezerwacja full outer join projekt.Klient  
on projekt.rezerwacja.klient_id = projekt.klient.klient_id where kurs_id=1;
```

LISTA KURSÓW DLA KONKRETNEGO KIEROWCY NA NAJBLIŻSZY TYDZIEŃ

```
select kurs_id, marka, trasa_start, godzina_wyjazdu, trasa_meta, godzina_przyjazdu from  
projekt.Kurs, projekt.Trasa, projekt.Pojazd where projekt.Kurs.trasa_id = projekt.Trasa.trasa_id and  
projekt.Kurs.pojazd_id = projekt.Pojazd.pojazd_id  
and projekt.Kurs.kierowca_id = 156801 and godzina_wyjazdu between '2017-01-08' and '2017-01-  
15';
```

PRZEPRACOWANE GODZINY KIEROWCÓW W KONKRETNYM TYGODNIU

```
select kierowca_id, sum(godzina_przyjazdu - godzina_wyjazdu) from projekt.kurs where  
kierowca_id = 1 and godzina_przyjazdu between '2017-01-07 00:00' and '2017-01-10 00:00'  
group by kierowca_id;
```

PRZEWIDZIANE ZAROBKI Z KONKRETNEGO KURSU LUB Z TYGODNIA NADCHODZĄCEGO

```
select r.kurs_id, k.godzina_wyjazdu, sum(r.liczba_miejsc*t.cena) from projekt.rezerwacja r,  
projekt.trasa t, projekt.kurs k where t.trasa_id = r.trasa_id and k.kurs_id = r.kurs_id and  
r.kurs_id=1 group by r.kurs_id, k.godzina_wyjazdu;
```

```
select r.kurs_id, k.godzina_wyjazdu, sum(r.liczba_miejsc*t.cena) from projekt.rezerwacja r,  
projekt.trasa t, projekt.kurs k where t.trasa_id = r.trasa_id and k.kurs_id = r.kurs_id and  
k.godzina_wyjazdu between '2017-01-08 00:00' and '2017-01-15 00:00' group by r.kurs_id,  
k.godzina_wyjazdu;
```

LISTA WYPOŻYCZEŃ NA KONKRETNY TYDZIEŃ LUB NA NAJBLIŻSZY

```
select k.imie, k.nazwisko, p.marka, w.data, kl.imie, kl.nazwisko from projekt.Pojazd p,  
projekt.Kierowca k, projekt.Wypozyczenie w, projekt.Klient kl where kl.klient_id = w.klient_id and  
k.kierowca_id = w.kierowca_id and p.pojazd_id = w.pojazd_id  
and w.data between '$start' and '$end';
```

RAPORT O STANIE POJAZDÓW

```
select pojazd_id, marka, przeglad, pozyczony from projekt.Pojazd;
```

IŁOŚĆ TRAS POKONANYCH PRZEZ POJAZDY DO TEJ PORY

```
select projekt.kurs.pojazd_id, marka, count(*) from projekt.Pojazd right join projekt.Kurs on  
projekt.Kurs.pojazd_id=projekt.Pojazd.pojazd_id group by projekt.Pojazd.marka,  
projekt.kurs.pojazd_id;
```

LISTA PROMOCJI PRZYPISANYCH KLIENTOM LUB KONKRETNEMU KLIENTOWI

```
select k.klient_id, k.imie, k.nazwisko, k.Promocja_ID, p.nazwa from projekt.Klient k,  
projekt.Promocja p where p.promocja_id = k.promocja_id;
```

```
select k.klient_id, k.imie, k.nazwisko, k.Promocja_ID, p.nazwa from projekt.Klient k,  
projekt.Promocja p where p.promocja_id = k.promocja_id and k.klient_id = $id;
```

IŁOŚĆ TRAS POKONANYCH PRZEZ KLIENTÓW LUB PRZEZ KONKRETNEGO KLIENTA

```
select k.klient_id, k.imie, k.nazwisko, count(*) from projekt.Klient k, projekt.Rezerwacja r where  
k.klient_id = r.klient_id group by k.klient_id;
```

```
select k.klient_id, k.imie, k.nazwisko, count(*) from projekt.Klient k, projekt.Rezerwacja r where  
k.klient_id = r.klient_id and k.klient_id = 1 group by k.klient_id;
```

Przegląd danych:

USUWANIE CAŁEGO WIERSZA Z TABLICY PRZEZ POBRANIE ID

```
delete from projekt.klient where klient_id=1;

delete from projekt.pojazd where pojazd_id=256800;

delete from projekt.trasa where trasa_id=10;

delete from projekt.kurs where kurs_id=12;

delete from projekt.promocja where promocja_id=101;

delete from projekt.rezerwacja where rezerwacja_id = 1;

delete from projekt.wypozyczenie where wypozyczenie_id = 1;
```

DODAWANIE NOWEGO WIERSZA

```
insert into projekt.kierowca (Imie, Nazwisko, Pesel, Mail, Data_Zatrudnienia) values (...);

insert into projekt.pojazd (Marka, Data_Produkcji, Data_Zakupu, Liczba_Miejsc, Przegląd,
Cena_Wypozyczenia, Pozyczony) values (...);

insert into projekt.trasa (Trasa_Start, Trasa_Meta, Odleglosc, cena) values (...);

insert into projekt.promocja (Nazwa, Darmowe_Przejazdy) values ('$info1', $info2);

insert into projekt.klient (Imie, Nazwisko, Pesel, Mail, Haslo, Telefon) values (...);

insert into projekt.kurs (Pojazd_ID, Kierowca_ID, Trasa_Id, Liczba_Pasazerow, Godzina_Wyjazdu,
Godzina_Przyjazdu) values (...);

insert into projekt.rezerwacja (Klient_ID, Pojazd_ID, Kierowca_ID, Data) values (...)

insert into projekt.wypozyczenie (Klient_ID, Trasa_Id, Kierowca_ID, Pojazd_ID, Kurs_ID,
Liczba_Miejsc) values (...);
```

PRZEGLĄD DANYCH

```
select * from projekt.klient;

select * from projekt.pojazd;

select * from projekt.trasa;

select * from projekt.kurs;

select * from projekt.promocja;

select * from projekt.rezerwacja;

select * from projekt.wypozyczenie;
```

ZMIANA KONKRETNEJ DANEJ ZA PODANIEM ID

```
update projekt.klient set zmienna = wartosc where kierowca_id = konkretnyID;

update projekt.pojazd set zmienna = wartosc where kierowca_id = konkretnyID;

update projekt.trasa set zmienna = wartosc where kierowca_id = konkretnyID;

update projekt.kurs set zmienna = wartosc where kierowca_id = konkretnyID;

update projekt.promocja set zmienna = wartosc where kierowca_id = konkretnyID;

update projekt.rezerwacja set zmienna = wartosc where kierowca_id = konkretnyID;

update projekt.wypozyczenie set zmienna = wartosc where kierowca_id = konkretnyID;
```

4 Projekt funkcjonalny

4.1 Interfejsy do prezentacji, edycji i obsługi danych

Strona główna prezentuje zapisane kursy na dzisiejszy dzień. W panelu *menu* można znaleźć zakładki:

- **Strefa pracownika**
- **Strefa klienta**
- **Przegląd danych**
- **Edycja danych**

Każda z zakładek posiada na swojej stronie osobne menu, gdzie może wybrać potrzebny raport lub tablicę do zbadania.

W zakładce **Strefa pracownika** zawarto dużą ilość raportów.

- **Dane najbliższe kursów**

Nastąpi automatyczne wyświetlenie danych kursów na dzień dzisiejszy.

- **Rezerwacje na kursy**

Zakładka umożliwia pobranie danych klientów, którzy dokonali rezerwacji na konkretny kurs. Należy najpierw podać *ID* kursu np. 13, a następnie kliknąć *Sprawd.* Wyświetli się tabela z imieniem, nazwiskiem i liczbą zarezerwowanych miejsc.

- **Kursy kierowców**

Zakładka pozwala na pobranie rozpiski kursów na cały tydzień licząc od daty sprawdzenia raportu dla konkretnego kierowcy. Należy najpierw podać identyfikator kierowcy np. 100000, a następnie wcisnąć *Sprawd.* Wyświetli się *ID* kursu, marka przypisanego pojazdu, miejsca przystanków oraz godziny wyjazdów i przyjazdów.

- **Kierowcy - godziny** Zakładka umożliwia zbadanie działalności kierowców, ich częstość pracy. Po wpisaniu identyfikatora pracownika wyświetli się *ID* kierowcy oraz przepracowane godziny w danym miesiącu.

- **Przewidziany zarobek**

Automatycznie wyświetli się tablica kursów z *ID* kursu, datą i godziną wyjazdu oraz przewidywanym zarobkiem z najbliższych 7 dni.

Po podaniu *ID* kursu np.14 wyświetli się zarobek z konkretnego kursu.

- **Lista wypożyczeń**

Automatycznie wyświetli się tablica wypożyczeń zanotowanych na najbliższy tydzień.

Po wpisaniu konkretnych dat jako ram czasowych, wyświetli się tablica danych z danego okresu.

Tablica danych przedstawia się następująco: Imię kierowcy, Nazwisko kierowcy, Marka pojazdu, Data, Imię klienta, Nazwisko klienta.

- **Stan pojazdów**

Zostanie wyświetlona tablica z *ID* pojazdu, marka samochodu, data przeglądu i stan czy pojazd czy pojazd jest wypożyczony czy nie. Jeśli jest lub kursuje, flaga wynosi *f*.

- **Ilość tras pojazdów**

Wyświetlimy tu tablicę informacji o eksploatacji pojazdu: ile tras do tej pory pojazd przejechał. Tablica wygląda następująco: ID pojazdu, marka, ilość wykonanych tras.

W zakładce **Strefa klienta** znajdziemy dwie opcje: **Ilość tras klientów**:

zostanie wyświetlona tablica wszystkich klientów i wykonanych przez nich tras, zaś po wpisaniu identyfikatora uzyskamy informacje o konkretnej osobie;

oraz **Promocja dla klientów**: Otrzymamy tablicę wszystkich klientów z uzyskanymi przez nich promocjami. Po wpisaniu identyfikatora otrzymamy obraz konkretnego klienta.

Opcja **Edytuj** pozwala na edycję konkretnego wiersza w konkretnej tablicy.

Należy tutaj najpierw podać identyfikator wiersza, w którym chcemy dokonać zmian. Następnie wprowadzamy zmienną i potwierdzamy przyciskiem obok.

Jeżeli dany identyfikator istnieje w tablicy, operacja się powiedzie. Należy jednak sprawdzić, czy operacja nie została odrzucona przez wbudowany mechanizm walidacji danych w bazie PostgreSQL. Wiersz możemy sprawdzić w zakładce **Przegląd danych**.

Można również usunąć cały wiersz lub dodać nowy wiersz.

W celu usunięcia należy w menu zakładki **Edycja danych** znaleźć opcję **Usuń**, następnie wprowadzić odpowiednie ID wiersza i potwierdzić przyciskiem niżej.

Opcję dodawania nowego wiersza znajdziemy w **Dodaj**. Należy wprowadzić dane właściwego typu. Warto tutaj zaznajomić się ze słownikiem danych. Zwracam uwagę na wprowadzanie dat, zwłaszcza w tabeli *kursw*. Operację należy potwierdzić przyciskiem, a następnie sprawdzić poprawne wykonanie danych w **Przeglądzie danych**.

4.2 Wizualizacja danych

Dane są przedstawione w odpowiednich tabelkach. Wyświetlane są automatycznie, lub dopiero po wprowadzeniu odpowiednich identyfikatorów.

5 Dokumentacja

5.1 Wprowadzenie danych

Dane należy wprowadzać ręcznie z uwzględnieniem odpowiedniego typu danych i ich ograniczeń.

5.2 Dokumentacja użytkownika

Chcąc się połączyć ze stroną, należy najpierw zalogować się na serwer *taurus.fis.agh.edu.pl* lub *pascal.fis.agh.edu.pl*. Następnie włączamy przeglądarkę np. Firefox.

Adres do wywołania strony:

http : //pascal.fis.agh.edu.pl/ 4holik/BAZA/index.php

5.3 Opracowanie dokumentacji technicznej

Program został utworzony w języku PHP jako strona internetowa na serwerze *pascal*. Zastosowano również style CSS. Wykorzystuje tamtejszą bazę PostgreSQL. Baza zawarta jest w schemacie *projekt* na koncie użytkownika *u4holik*.

Dokumentacja techniczna znajduje się w plikach *.php* dołączonych do dokumentacji.

5.4 Wykaz literatury

<https://neobus.pl/>

<http://www.vdlbuscenter.com/>

<http://php.net/manual/pl/book.pgsql.php>