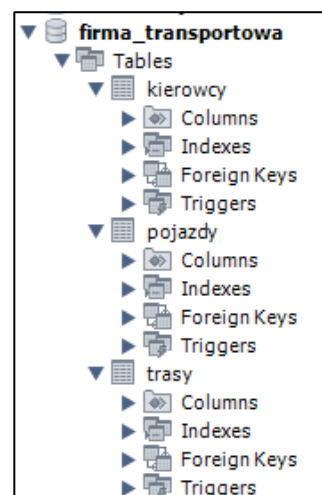


Wydział WIMIIP	Imię i nazwisko 1.Kacper Bielak 2.Tomasz Artecki 3.Dominik Budzowski	Rok 2	Grupa GL01	
Temat: [CWP1] Projekt MySQL				
Data wykonania 16.05.2021	Bazy danych	Przedmiot:	OCENA	

## 1. Idea projektu

Baza danych służąca do zarządzania firmą transportową. Zawiera tabele pojazdów, tras oraz kierowców. Umożliwia dodawanie i usuwanie rekordów za pomocą procedur. Użytkownik może z łatwością selekcjonować oraz wyświetlać najpotrzebniejsze dane za pomocą widoków i innych narzędzi. Projekt bazy jest rozwinięciem projektu bazy danych z MS Access.



## 2. Diagram ERD z uzasadnieniem relacji (Może to być z MS Access, może być narysowany np. w Visio/Lucidchart, draw.io)

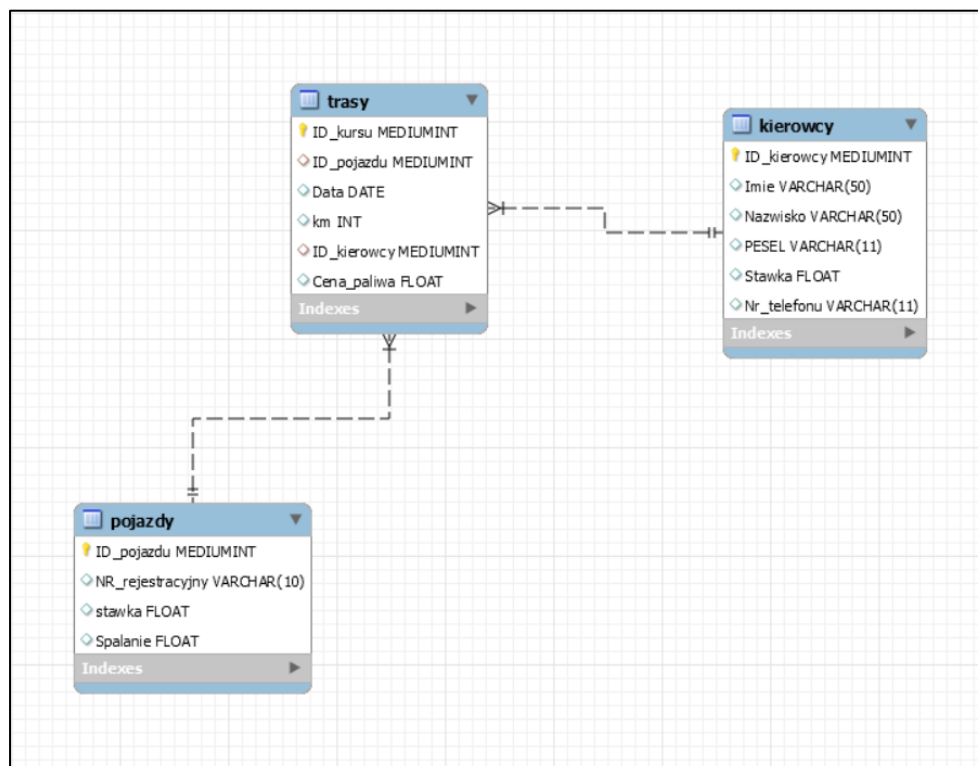


Tabela trasy jest powiązana z tabelą pojazdy poprzez ID pojazdu oraz z tabelą kierowcy poprzez ID kierowcy. Relacje te są typu jeden do wielu.

### 3. SQL DCL – utworzenie użytkowników i nadanie im uprawnień

W bazie utworzono 2 użytkowników, jednym z nich jest właściciel, który ma wszystkie uprawnienia do całej bazy, a drugim jest pracownik, które może operować tylko na tabeli *trasy*:

```
create user wlasciciel identified by 'wlasciciel';
Grant all privileges on * to wlasciciel;
create user pracownik identified by 'pracownik';
Grant all privileges on trasy.* to pracownik;
```

### 4. Złączenia poziome (SQL JOIN) oraz złączenia pionowe (SQL UNION, UNION ALL INTERSECT, EXCEPT)

Projekt zawiera liczne złączenia poziome i pionowe, przykłady:

#### **Wykaz tras pojazdów (złączenie pionowe)**

```
create view wykaz_tras_pojazdow as
SELECT Pojazdy.ID_pojazdu, Pojazdy.NR_rejestracyjny, Trasy.Data, Sum(Trasy.km) AS
Suma_kilometrów
FROM Pojazdy INNER JOIN (Kierowcy INNER JOIN Trasy ON
Kierowcy.ID_kierowcy = Trasy.ID_kierowcy) ON Pojazdy.ID_pojazdu = Trasy.ID_pojazdu
GROUP BY Pojazdy.ID_pojazdu, Pojazdy.NR_rejestracyjny, Trasy.Data ORDER BY
Trasy.Data DESC;
```

#### **Pokaz trasy kierowcy danym pojazdem (złączenie poziome)**

```
delimiter $
create procedure pokaz_trasy_kierowcy_danym_pojazdem(
IN id_kier mediumint, IN id_poj mediumint)
begin
select id_kierowcy as id, stawka as stawka_i_cenapaliwa
from kierowcy where id_kierowcy=id_kier union
all select id_pojazdu, stawka from pojazdy
where id_pojazdu=id_poj union all select id_kursu, Cena_paliwa from trasy
where ID_kierowcy=id_kier and ID_pojazdu=id_poj;
end $
delimiter ;
```

### 5. Procedury, Wyzwalacze, Widoki

#### **a) Procedury**

Projekt zawiera 10 procedur, które odpowiadają za dodawanie i usuwanie rekordów do/z tabel, wyświetlenie tras danego kierowcy danym pojazdem, zarobki danego pojazdu, pracownika oraz zestawienie zarobków pracowników w danym miesiącu:



## b) Wyzwalacze

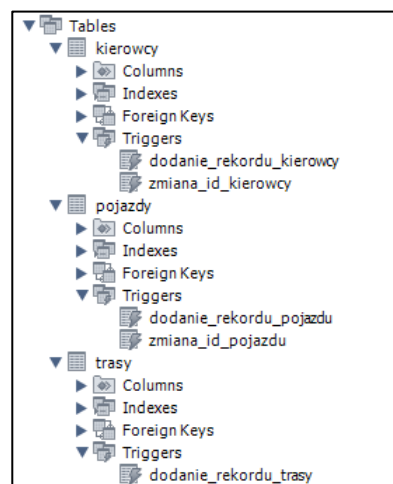
Projekt zawiera 5 wyzwalaczy, które odpowiadają za pilnowanie wstawiania poprawnych danych do nowych rekordów oraz równoległą aktualizację ID\_pojazdu i ID\_kierowcy w tabeli trasy przy zmianie tych danych w tabelach kierowcy i pojazdy:

(Before)

- dodanie\_rekordu\_kierowcy
- dodanie\_rekordu\_pojazdu
- dodanie\_rekordu\_trasy

(After)

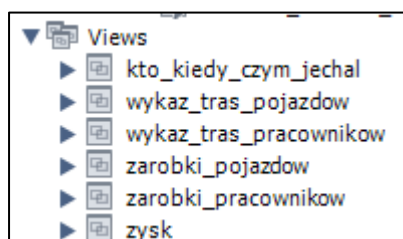
- zmiana\_id\_kierowcy
- zmiana\_id\_pojazdu



## c) Widoki

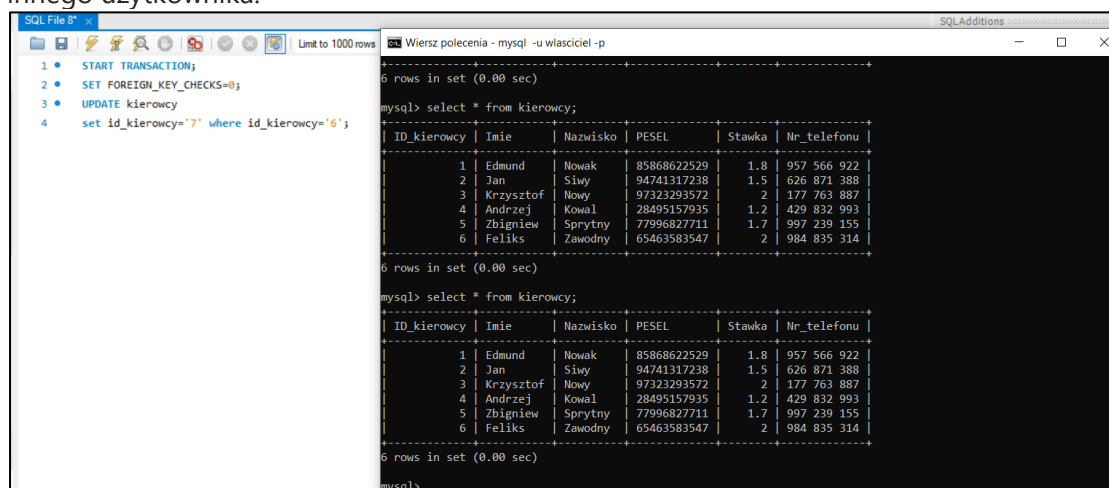
Projekt zawiera 6 widoków:

- **kto\_kiedy\_czym\_jechal** - wyświetla informacje o kursie zawierające dane kierowcy, pojazdu oraz datę;
- **wykaz\_tras\_pojazdow** - wyświetla liczbę kilometrów danego pojazdu w danym dniu;
- **wykaz\_tras\_pracownikow** - wyświetla liczbę kilometrów danego kierowcy w danym dniu;
- **zarobki\_pojazdow** - wyświetla dane finansowe – zarobki, wydatki na paliwo oraz zarobki na czysto danego pojazdu w danym miesiącu oraz sumę przejechanych kilometrów;
- **zarobki\_pracownikow** – wyświetla dane finansowe pracowników w danym miesiącu oraz sumę przejechanych kilometrów;
- **zysk** - wyświetla dane finansowe firmy – zarobki pojazdów, pracowników oraz czysty zysk w danym miesiącu;



## 6. Wykorzystanie transakcji i pokazanie że działa

Poniżej znajduje się przykładowa transakcja, w workbenchu z perspektywy admina i w cmd z perspektywy innego użytkownika:



Rozpoczęcie transakcji i wykonanie aktualizacji rekordu bez jej zakończenia. Można zauważyć brak jakiegokolwiek zmiany z perspektywy użytkownika w cmd.

The screenshot shows a SQL File editor on the left with the following script:

```

1 • START TRANSACTION;
2 • SET FOREIGN_KEY_CHECKS=0;
3 • UPDATE kierowcy
4   set id_kierowcy='7' where id_kierowcy='6';
5 • rollback;

```

The MySQL terminal on the right shows the execution of the script:

```

mysql> select * from kierowcy;
+----+-----+-----+-----+-----+-----+
| ID_kierowcy | Imie   | Nazwisko | PESEL   | Stawka | Nr_telefonu |
+----+-----+-----+-----+-----+-----+
| 1 | Edmund | Nowak   | 85868622529 | 1.8 | 957 566 922 |
| 2 | Jan    | Siwy    | 94741317238 | 1.5 | 626 871 388 |
| 3 | Krzysztof | Nowy  | 97323293572 | 2 | 177 763 887 |
| 4 | Andrzej | Kowal   | 28495157935 | 1.2 | 429 832 993 |
| 5 | Zbigniew | Sprytny | 77996827711 | 1.7 | 997 239 155 |
| 6 | Feliks  | Zawodny | 65463583547 | 2 | 984 835 314 |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from kierowcy;
+----+-----+-----+-----+-----+-----+
| ID_kierowcy | Imie   | Nazwisko | PESEL   | Stawka | Nr_telefonu |
+----+-----+-----+-----+-----+-----+
| 1 | Edmund | Nowak   | 85868622529 | 1.8 | 957 566 922 |
| 2 | Jan    | Siwy    | 94741317238 | 1.5 | 626 871 388 |
| 3 | Krzysztof | Nowy  | 97323293572 | 2 | 177 763 887 |
| 4 | Andrzej | Kowal   | 28495157935 | 1.2 | 429 832 993 |
| 5 | Zbigniew | Sprytny | 77996827711 | 1.7 | 997 239 155 |
| 6 | Feliks  | Zawodny | 65463583547 | 2 | 984 835 314 |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

Rozpoczęcie transakcji, wykonanie aktualizacji rekordu oraz odrzucenie zmian. Efekt w cmd jak w powyższym przypadku.

The screenshot shows a SQL File editor on the left with the following script:

```

1 • START TRANSACTION;
2 • SET FOREIGN_KEY_CHECKS=0;
3 • UPDATE kierowcy
4   set id_kierowcy='7' where id_kierowcy='6';
5 • commit;

```

The MySQL terminal on the right shows the execution of the script:

```

mysql> select * from kierowcy;
+----+-----+-----+-----+-----+-----+
| ID_kierowcy | Imie   | Nazwisko | PESEL   | Stawka | Nr_telefonu |
+----+-----+-----+-----+-----+-----+
| 1 | Edmund | Nowak   | 85868622529 | 1.8 | 957 566 922 |
| 2 | Jan    | Siwy    | 94741317238 | 1.5 | 626 871 388 |
| 3 | Krzysztof | Nowy  | 97323293572 | 2 | 177 763 887 |
| 4 | Andrzej | Kowal   | 28495157935 | 1.2 | 429 832 993 |
| 5 | Zbigniew | Sprytny | 77996827711 | 1.7 | 997 239 155 |
| 6 | Feliks  | Zawodny | 65463583547 | 2 | 984 835 314 |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from kierowcy;
+----+-----+-----+-----+-----+-----+
| ID_kierowcy | Imie   | Nazwisko | PESEL   | Stawka | Nr_telefonu |
+----+-----+-----+-----+-----+-----+
| 1 | Edmund | Nowak   | 85868622529 | 1.8 | 957 566 922 |
| 2 | Jan    | Siwy    | 94741317238 | 1.5 | 626 871 388 |
| 3 | Krzysztof | Nowy  | 97323293572 | 2 | 177 763 887 |
| 4 | Andrzej | Kowal   | 28495157935 | 1.2 | 429 832 993 |
| 5 | Zbigniew | Sprytny | 77996827711 | 1.7 | 997 239 155 |
| 7 | Feliks  | Zawodny | 65463583547 | 2 | 984 835 314 |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

Rozpoczęcie transakcji, wykonanie aktualizacji rekordu oraz zaakceptowanie zmian. W efekcie widzimy aktualizację na innym użytkowniku w cmd.

Ponadto transakcje zostały użyte przy pisaniu CRUDa co będzie opisane w kolejnej części sprawozdania.

## 7. CRUD

CRUD został napisany w języku C++ w środowisku Visual Studio 2019 korzystając z bibliotek MySQL Server 8.0 oraz MySQL Connector C++ 1.1.13. Program łączy się z bazą danych Firmy Transportowej i udostępnia przejrzysty interfejs konsolowy do jej zarządzania. Program opiera się na działaniu transakcji, dzięki czemu można odrzucać lub zatwierdzać zmiany.

**Link do repozytorium:**

<https://github.com/DominikBu/CRUD>

### Działanie programu

Na początku dostajemy informacje czy połączono i podajemy dla jakiej tabeli będą wykonywane operacje.

```

Polaczono!

Podaj na jakiej tabeli chcesz operowac
[1] kierowcy
[2] pojazdy
[3] trasy

```

Nazwa tabeli, którą wybraliśmy będzie widoczna cały czas w menu.

Po podaniu wyświetla się nam menu z różnymi opcjami.

```
Firma transportowa
Aktualna tabela: kierowcy
Wybierz opcje:
[1] Wyświetl tabele
[2] Dodaj
[3] Odczytaj
[4] Edytuj
[5] Usun
[6] Zmien tabele
[7] Zatwierdz zmiany
[8] Anuluj zmiany
[0] Wyjdz
```

Po wpisaniu odpowiedniej cyfry program wykonuje pewne operacje, a po wykonaniu ich wyświetla ponownie menu.

**1. Wyświetl tabelę** – wyświetla tabelę, a dokładnie nazwy kolumn i wszystkie jej wartości. Wyświetlanie w standardowej konsoli c++ nie uwzględnia prawidłowo wszystkich znaków np. ś, ć, ó itd.

*Fragment wyniku z konsoli:*

```
1
ID_kierowcy   Imie   Nazwisko   PESEL   Stawka   Nr_telefonu
1      Edmund   Nowak   85868622529   1.8     957 566 922
2      Jan     Siwy    94741317238   1.5     626 871 388
3      Krzysztof   Nowy    97323293572   2       177 763 887
4      Andrzej   Kowal   28495157935   1.2     429 832 993
5      Zbigniew   Sprytny 77996827711   1.7     997 239 155
6      Feliks   Zawodny 65463583547   2       984 835 314
```

**2. Dodaj** – dodaje element to tabeli, musimy podać wartości każdej kolumny.

```
2
Podaj wartosc pola: ID_kierowcy
7
Podaj wartosc pola: Imie
Dominik
Podaj wartosc pola: Nazwisko
Guard
Podaj wartosc pola: PESEL
83928378932
Podaj wartosc pola: Stawka
2.1
Podaj wartosc pola: Nr_telefonu
897 876 789
Udalo sie dodac element!
```

Po wypisaniu tabeli doda nam się wiersz.

```
7      Dominik   Guard   83928378932   2.1     897 876 789
```

**3. Odczytaj** – odczytuje wartość pola we wskazanym miejscu w tabeli.

```
3
Podaj kolumne oraz wiersz dla ktorej chcesz uzyskac dane
Ilosc kolumn: 6
Ilosc wierszy: 6
3
2
Kolumna: Nazwisko
Wartosc: Siwy
```

**4. Edytuj** – edytuje wartość pola we wskazanym miejscu tabeli na podaną wartość

```
4
Podaj kolumne oraz wiersz dla ktorej chcesz uzyskac dane i je zmienic:
Ilosc kolumn: 6
Ilosc wierszy: 6
3
2
Kolumna: Nazwisko
Wartosc: Siwy
Podaj wartosc na jaka chcesz podmienic:
Niebieski
Udalo sie edytowac element!
```

Znowu mogą wystąpić problemy gdy podamy niewłaściwą wartość w przypadku kluczy.

*Przykład:*

```
4
Podaj kolumne oraz wiersz dla ktorej chcesz uzyskac dane i je zmienic:
Ilosc kolumn: 6
Ilosc wierszy: 6
1
3
Kolumna: ID_kierowcy
Wartosc: 3
Podaj wartosc na jaka chcesz podmienic:
4
Nie udalo sie edytowac elementu!
```

**5. Usun** – Usuwa podany wiersz tabeli.

```
5
Podaj wiersz ktory chcesz usunac:
Ilosc wierszy: 6
6
ID_kierowcy      Imie      Nazwisko      PESEL      Stawka      Nr_telefonu
6      Feliks  Zawodny 65463583547      2      984 835 314
Udalo sie usunac element!
```

**6. Zmien table** – prosi o podanie nazwy tabeli na której będziemy teraz pracować.

```
6
Podaj na jakiej tabeli chcesz operowac
[1] kierowcy
[2] pojazdy
[3] trasy
```

I teraz po wybraniu opcji 1 wyświetli się:

```

Firma transportowa
Aktualna tabela: pojazdy
Wybierz opcje:
[1] Wyszwietl tabele
[2] Dodaj
[3] Odczytaj
[4] Edytuj
[5] Usun
[6] Zmien tabele
[7] Zatwierdz zmiany
[8] Anuluj zmiany
[0] Wyjdz

```

**7. Zatwierdz zmiany** – zatwierdza i zapisuje zmiany poleceniem COMMIT co spowoduje, że gdy wyjdziemy z programu i wrócimy na nowo zmiany tam będą. Po zapisie zmian program rozpoczyna nową instrukcję START TRANSACTION.

```

7
Zatwierdzono zmiany

```

### 8. Anuluj zmiany

```

8
Anulowano zmiany

```

**0. Wyjdz** – Kończy działanie programu.

## 8. Wnioski

MySQL jest świetnym środowiskiem bazodanowym, które bardzo pomaga w tworzeniu i zarządzaniu bazami danych. Szczególnie MySQL Workbench jest bardzo intuicyjny i łatwy w obsłudze. Wszystkie elementy baz danych takie jak tabele, widoki, procedury, wyzwalacze są w przejrzysty sposób wyświetlone w lewej części okna programu, dzięki czemu można w łatwy sposób przemieszczać się między nimi. Nie bez powodu użyteczność MySQL potwierdzają nawet rankingi popularności użytkowników baz danych. Do napisania CRUDa wybraliśmy właśnie C++, ponieważ jest on zintegrowany z bardzo użyteczną biblioteką do łączenia i zarządzania bazą danych. Dzięki połączeniu bazy danych z językiem programowania możemy w łatwy sposób korzystać z danych, wykonywać na nich operacje, oraz jeśli jest taka konieczność to usuwać, aktualizować bądź zapisywać nowe dane. Połączenie z bazą danych można traktować jako nowy etap gdzie już nie jesteśmy zmuszeni operować z danymi pobranymi z notatnika. W porównaniu ze środowiskiem MS Access praca w MySQL jest dużo bardziej uniwersalna, dzięki możliwości eksportowania danych do środowisk opierających się o język SQL, oraz bardziej efektywna z powodu większych możliwości rozbudowy bazy i dużo łatwiejszej obsługi w przypadku większych zbiorów danych.