

Projekt zaliczeniowy JDBC

Norbert Dąb lab. 1

1. Temat projektu

Aplikacja desktopowa napisana w Javie umożliwiająca łączenie się z zaprojektowaną i utworzoną uprzednio relacyjną bazą danych oraz jej obsługę.

Projekt napisany jest w technologii JavaFX+JDBC.

Jak serwer bazy danych wykorzystuję MySQL.

2. Autor

Norbert Drąg

Laboratorium 1

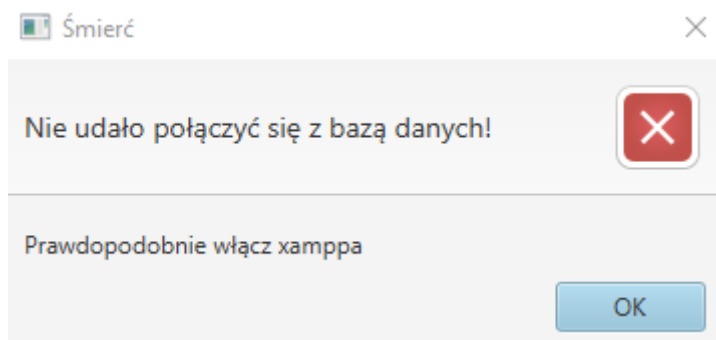
Nr albumu 117790

3. Link do GitHub

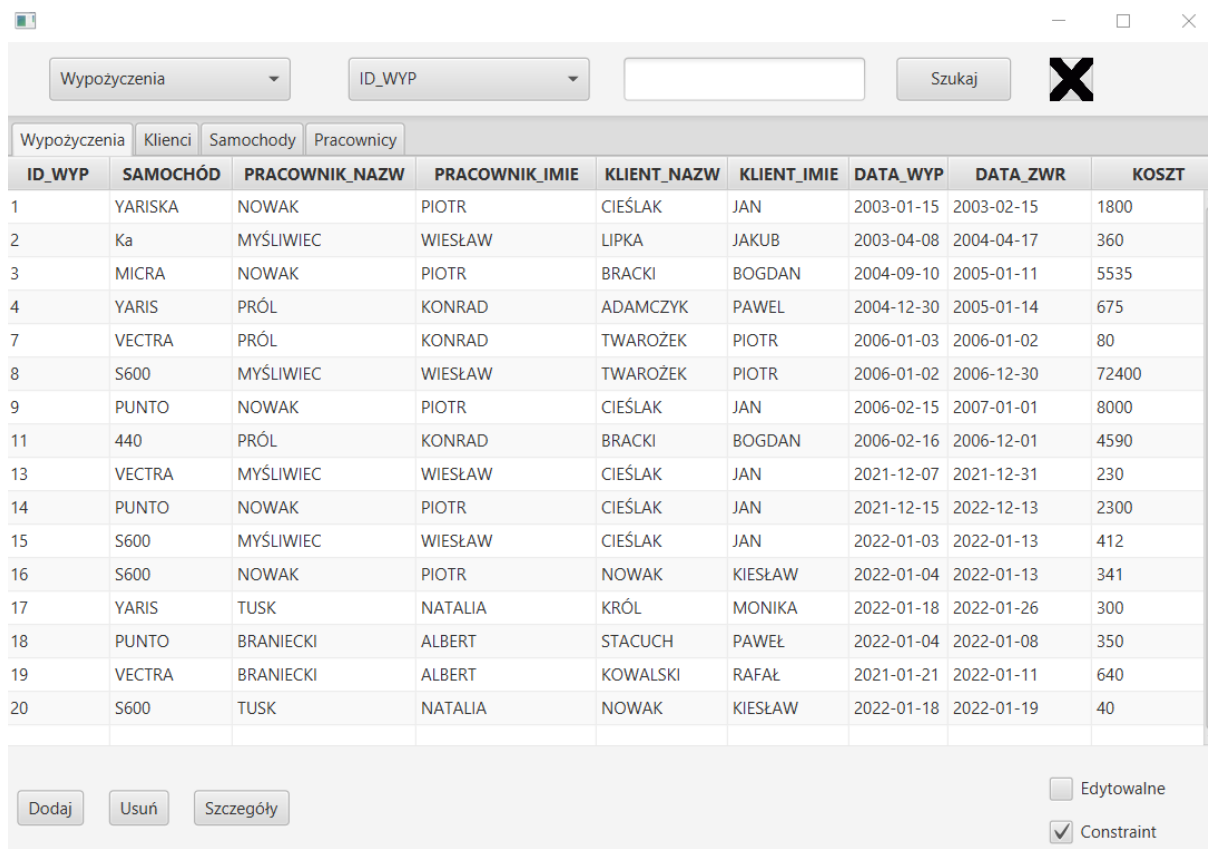
Link do githuba:

<https://github.com/ZurisW/Projekt-JDBC>

4. Wytlumaczenie działania aplikacji



Gdy nie można połączyć się z bazą danych wyskakuje okienko erroru.



Główny widok aplikacji, w którym można wyszukiwać po danych kolumnach, dodawać i usuwać z bazy danych oraz pokazywać szczegóły. Ponadto są dwa checkboxy, które kontrolują możliwość edytowania bazy danych oraz włączenia/wyłączenia sprawdzania kluczy obcych.

Widok zrobiony jest na TabPane, by szybko przełączać się między kolumnami bazy danych. Reszta kolumn:

Klienci:

Klienci

ID_KLI

Wypożyczenia

Klienci

Samochody

Pracownicy

ID_KLI	NAZWISKO	IMIE	NR_DOWODU	MIEJSCOWOSC	ULICA
10	TWAROŻEK	PIOTR	AA1234567	WARSZAWA	DLUGA 8/12
15	CIEŚLAK	JAN	BC2343561	WROCLAW	NORWIDA 2/1
20	ADAMCZYK	PAWEL	AG8967452	POZNAN	KROTKA 2
25	BRACKI	BOGDAN	CC3478690	GDANSK	BALTYCKA 67/4
30	LIPKA	JAKUB	CE6712098	KATOWICE	POLNA 9
31	BOŻEK	PIOTR	SD3321643	KRYG	MELA 2/4
33	NOWAK	KIESŁAW	JD4280120	SZYMBARG	KAPIBARA 4/10
34	KAPECKI	ADRIAN	DW4817654	DĘBICA	RODOKOT 3/20
35	STACUCH	PAWEŁ	JK4001453	SĘKOWA	KAZIMIERZA 7/10
36	KOWALSKI	RAFAŁ	IE0731282	PIASECZNO	OGROWA 1/85
37	KRÓL	MONIKA	MK3060390	ŁAŃCUT	GŁÓWNA 3/36

Dodaj

Usuń

Szukaj

☐ Edytowalne

☒ Constraint

Samochody:

Samochody

ID_SAM

Wypożyczenia

Klienci

Samochody

Pracownicy

ID_SAM	NR_REJ	MARKA	MODEL	ROK_PROD	KRAJ_PROD	POJ_SIL	KOSZT_DNIA
100	KRA-1023	OPEL	ASTRA	1995	NIEMCY	1	45
110	KRC-A120	MERCEDES	S600	1998	NIEMCY	6	200
120	TAV-1909	FORD	Ka	2000	USA	1	40
130	WAW-8967	TOYOTA	YARIS	2001	JAPONIA	1	45
140	WRE-4509	OPEL	VECTRA	1997	NIEMCY	3	80
150	GDA-A890	OPEL	YARISKA	1995	NIEMCY	2	60
160	PKT-0967	VOLVO	440	1984	SZWECJA	1	30
170	TAW-6598	FIAT	PUNTO	1991	WLOCHY	1	40
180	KRC-4590	NISSAN	MICRA	1998	JAPONIA	1	45
190	POK-9089	OPEL	CALIBRA	1990	NIEMCY	2	55
200	HPJ-4205	MELAK	MELINIARZ	1995	NIGGERIA	3	59

Dodaj

Usuń

Szukaj

☐ Edytowalne

☒ Constraint

Pracownicy:

Pracownicy

ID_PRAC

Szukaj

X

Wypożyczenia

Klienci

Samochody

Pracownicy

ID_PRAC	NAZWISKO	IMIE	TELEFON	PESEL
1	NOWAK	PIOTR	234532123	32323242125
2	MYŚLIWIEC	WIESŁAW	842902136	94215329106
3	PRÓŁ	KONRAD	942150521	21252196224
5	RUBY	MATI	952734124	13295916278
6	BRANIECKI	ALBERT	492142189	2123953292
7	TUSK	NATALIA	492612429	1302485932

Dodaj

Usuń

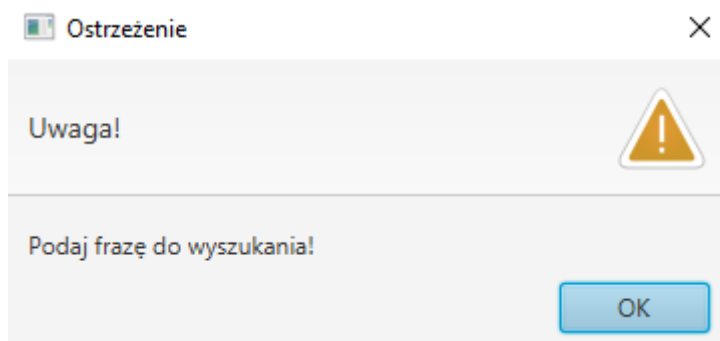
☐ Edytowalne

☒ Constraint

[illegible]

Pole oraz tabelę resetujemy guzikiem obok z grafiką X.

Jeśli nic nie wpisujemy dostaniemy następujące ostrzeżenie:



Edytowanie:

Edycja danych jest prosta i intuicyjna. Gdy włączymy checkboxa od edytowania (który domyślnie jest wyłączony), po podwójnym kliknięciu możemy zmienić dane rekordu.

Wypożyczenia

MODEL

Wypożyczenia

Klienci

Samochody

Pracownicy

ID_WYP	SAMOCHÓD	PRACOWNIK	KLIENT	DATA_WYP	DATA_ZWR	KOSZT
1	YARISKA	NOWAK	CIEŚLAK	2003-01-15	2003-02-15	1800
2	Ka	MYŚLIWIEC	LIPKA	2003-04-08	2004-04-17	360
3	MICRA	NOWAK	BRACKI	2004-09-10	2005-01-11	5535
4	YARIS	PRÓL	ADAMCZYK	2004-12-30	2005-01-14	675
7	VECTRA	PRÓL	TWAROŻEK	2006-01-03	2006-01-02	80
8	S600	MYŚLIWIEC	TWAROŻEK	2006-01-02	2006-12-30	72400
9	PUNTO	NOWAK	CIEŚLAK	2006-02-15	2007-01-01	8000
11	440	PRÓL	BRACKI	2006-02-16	2006-12-01	4590
13	VECTRA	MYŚLIWIEC	CIEŚLAK	2021-12-07	2021-12-31	230
14	PUNTO	NOWAK	CIEŚLAK	2021-12-15	2022-12-13	2300
15	S600	MYŚLIWIEC	CIEŚLAK	2022-01-03	2022-01-13	412
16	S600	NOWAK	NOWAK	2022-01-04	2022-01-13	341
17	YARIS	TUSK	KRÓL	2022-01-18	2022-01-26	300
18	PUNTO	BRANIECKI	STACUCH	2022-01-04	2022-01-08	350
19	VECTRA	BRANIECKI	KOWALSKI	2021-01-21	2022-01-11	640
20	S600	TUSK	NOWAK	2022-01-18	2022-01-19	40

Dodaj

Usuń

Szukaj

Szczegóły

☒ Edytowalne

☒ Constraint

Zmieniłem dla testu nazwisko pracownika Tusk na Jędrzej.

Wypożyczenia

MODEL

Wypożyczenia

Klienci

Samochody

Pracownicy

ID_WYP	SAMOCHÓD	PRACOWNIK	KLIENT	DATA_WYP	DATA_ZWR	KOSZT
1	YARISKA	NOWAK	CIEŚLAK	2003-01-15	2003-02-15	1800
2	Ka	MYŚLIWIEC	LIPKA	2003-04-08	2004-04-17	360
3	MICRA	NOWAK	BRACKI	2004-09-10	2005-01-11	5535
4	YARIS	PRÓL	ADAMCZYK	2004-12-30	2005-01-14	675
7	VECTRA	PRÓL	TWAROŻEK	2006-01-03	2006-01-02	80
8	S600	MYŚLIWIEC	TWAROŻEK	2006-01-02	2006-12-30	72400
9	PUNTO	NOWAK	CIEŚLAK	2006-02-15	2007-01-01	8000
11	440	PRÓL	BRACKI	2006-02-16	2006-12-01	4590
13	VECTRA	MYŚLIWIEC	CIEŚLAK	2021-12-07	2021-12-31	230
14	PUNTO	NOWAK	CIEŚLAK	2021-12-15	2022-12-13	2300
15	S600	MYŚLIWIEC	CIEŚLAK	2022-01-03	2022-01-13	412
16	S600	NOWAK	NOWAK	2022-01-04	2022-01-13	341
17	YARIS	JĘDRZEJ	KRÓL	2022-01-18	2022-01-26	300
18	PUNTO	BRANIECKI	STACUCH	2022-01-04	2022-01-08	350
19	VECTRA	BRANIECKI	KOWALSKI	2021-01-21	2022-01-11	640
20	S600	JĘDRZEJ	NOWAK	2022-01-18	2022-01-19	40

Dodaj

Usuń

Szukaj

Szczegóły

☒ Edytowalne
 ☒ Constraint

Oprócz zmiany w tabeli „Wypożyczenia” zmieniona została także tabela „Klienci”.

Pracownicy

ID_PRAC

Wypożyczenia

Klienci

Samochody

Pracownicy

ID_PRAC	NAZWISKO	IMIE	TELEFON	PESEL	
1	NOWAK	PIOTR	234532123	32323242125	
2	MYŚLIWIEC	WIESŁAW	842902136	94215329106	
3	PRÓL	KONRAD	942150521	21252196224	
5	RUBY	MATI	952734124	13295916278	
6	BRANIECKI	ALBERT	492142189	2123953292	
7	JĘDRZEJ	NATALIA	492612429	1302485932	

Dodaj

Usuń

Szukaj

☒ Edytowalne
 ☒ Constraint

Jeśli postanowimy uzupełnić pole niczym to pokazuje się ostrzeżenie:

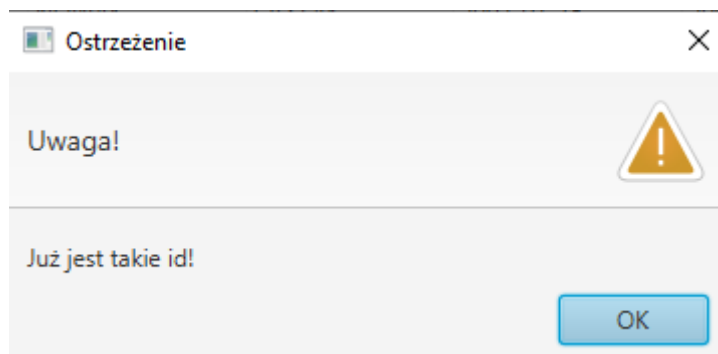
Ostrzeżenie

Uwaga!

Pole nie może być puste!

OK

Jeśli przy zmianie ID zmienimy na ID które już jest w bazie pojawia się następujące ostrzeżenie:



Dodawanie:

Aby dodać rekord do bazy danych naciskamy przycisk „Dodaj”. Następnie pojawia się okienko z wyborem tabeli do której chcemy wpisać dane oraz pola na nie.

Dodawanie do bazy danych

Wypożyczenia

SAMOCHÓD: Ka

PRACOWNIK: PRÓL KONRAD

KLIENT:

DATA_WYP:

DATA_ZWR:

KOSZT:

DO

BRANIECKI ALBERT

TUSK NATALIA

CIEŚLAK JAN

ADAMCZYK PAWEŁ

BRACKI BOGDAN

LIPKA JAKUB

BOŻEK PIOTR

NOWAK KIESŁAW

KAPECKI ADRIAN

STACUCH PAWEŁ

KOWALSKI RAFAŁ

KRÓL MONIKA

W tabeli wypożyczenia możemy wybrać po imieniu i nazwisku klienta oraz pracownika oraz model auta.

Dodawanie do bazy danych

Wypożyczenia

SAMOCHÓD:
Ka

PRACOWNIK:
PRÓŁ KONRAD

KLIENT:
KAPECKI ADRIAN

DATA_WYP:
02.02.2022

DATA_ZWR:

KOSZT:

DODAJ

BRANIECKI
ALBERT

TUSK
NATALIA


Luty
2022

	Pon.	Wt.	Śr.	Czw.	Pt.	Sob.	Niedz.
5	31	1	2	3	4	5	6
6	7	8	9	10	11	12	13
7	14	15	16	17	18	19	20
8	21	22	23	24	25	26	27
9	28	1	2	3	4	5	6
10	7	8	9	10	11	12	13

W polach data_wyp oraz data_zwr możemy wpisać datę bądź też wybrać ją z kalendarzyka. Wszystkie wpisywane dane są walidowane regexami na bieżąco oraz przy kliknięciu przycisku „DODAJ”. Textfieldy oraz labely pojawiają się i znikają przy wyborze odpowiedniej tabeli, przez co mogłem użyć jednego kontrolera. Dodany rekord:

22	Ka	PRÓŁ	KONRAD	KAPECKI	ADRIAN	2022-02-02	2022-02-26	350
----	----	------	--------	---------	--------	------------	------------	-----

Widok innych tabel:

 Dodawanie do bazy danych — □ ×

Klienci ▼

NAZWISKO:


IMIE:

NR_DOWODU:

MIEJSCOWOSC:

ULICA:

DODAJ

 Dodawanie do bazy danych — □ ×

Pracownicy ▼

NAZWISKO:

IMIE:

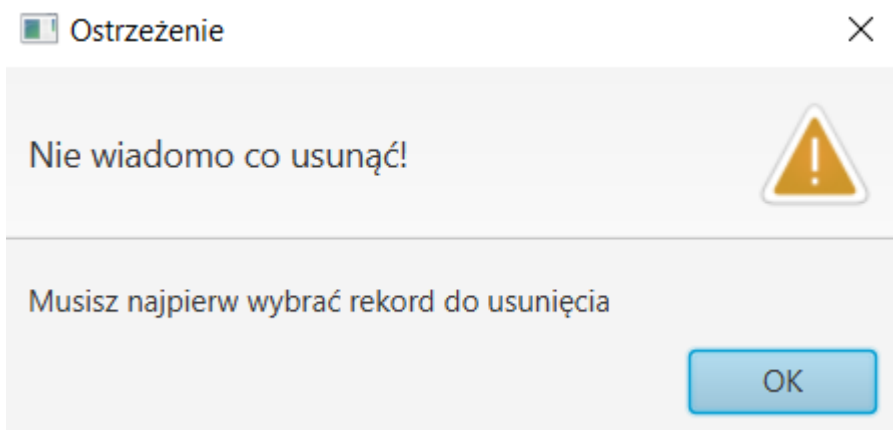
TELEFON:

PESEL:

DODAJ

Usuwanie:

Usuwanie rekordów z tablic jest bardzo proste. Wybieramy rekord oraz klikamy usuń. Jeśli jednak nie wybierzemy żadnego rekordu pokaże się nam ostrzeżenie:



Dla testu usunę rekord związany z panem Stacuchem.

ID_WYP	SAMOCHÓD	PRACOWNIK_NAZW	PRACOWNIK_IMIE	KLIENT_NAZW	KLIENT_IMIE	DATA_WYP	DATA_ZWR	KOSZT
2	Ka	MYŚLIWIEC	WIESŁAW	LIPKA	JAKUB	2003-04-08	2004-04-17	300
3	MICRA	NOWAK	PIOTR	BRACKI	BOGDAN	2004-09-10	2005-01-11	5535
4	YARIS	PRÓŁ	KONRAD	ADAMCZYK	PAWEŁ	2004-12-30	2005-01-14	675
7	VECTRA	PRÓŁ	KONRAD	TWAROŻEK	PIOTR	2006-01-03	2006-01-02	80
8	S600	MYŚLIWIEC	WIESŁAW	TWAROŻEK	PIOTR	2006-01-02	2006-12-30	72400
9	PUNTO	NOWAK	PIOTR	CIEŚLAK	JAN	2006-02-15	2007-01-01	8000
11	440	PRÓŁ	KONRAD	BRACKI	BOGDAN	2006-02-16	2006-12-01	4590
13	VECTRA	MYŚLIWIEC	WIESŁAW	CIEŚLAK	JAN	2021-12-07	2021-12-31	230
14	PUNTO	NOWAK	PIOTR	CIEŚLAK	JAN	2021-12-15	2022-12-13	2300
15	S600	MYŚLIWIEC	WIESŁAW	CIEŚLAK	JAN	2022-01-03	2022-01-13	412
16	S600	NOWAK	PIOTR	NOWAK	KIESŁAW	2022-01-04	2022-01-13	341
17	YARIS	TUSK	NATALIA	KRÓŁ	MONIKA	2022-01-18	2022-01-26	300
18	PUNTO	BRANIECKI	ALBERT	STACUCH	PAWEŁ	2022-01-04	2022-01-08	350
19	VECTRA	BRANIECKI	ALBERT	KOWALSKI	RAFAŁ	2021-01-21	2022-01-11	640
20	S600	TUSK	NATALIA	NOWAK	KIESŁAW	2022-01-18	2022-01-19	40
21	Ka	BRANIECKI	ALBERT	STACUCH	PAWEŁ	2022-02-08	2022-02-26	352
22	Ka	PRÓŁ	KONRAD	KAPECKI	ADRIAN	2022-02-02	2022-02-26	350

Dodaj Usuń Szczegóły

☐ Edytowalne
☒ Constraint

Wypożyczenia

ID_WYP

Szukaj

X

Wypożyczenia

Klienci

Samochody

Pracownicy

ID_WYP	SAMOCHÓD	PRACOWNIK_NAZW	PRACOWNIK_IMIE	KLIENT_NAZW	KLIENT_IMIE	DATA_WYP	DATA_ZWR	KOSZT
1	YARISKA	NOWAK	PIOTR	CIEŚŁAK	JAN	2003-01-15	2003-02-15	1800
2	Ka	MYŚLIWIEC	WIESŁAW	LIPKA	JAKUB	2003-04-08	2004-04-17	360
3	MICRA	NOWAK	PIOTR	BRACKI	BOGDAN	2004-09-10	2005-01-11	5535
4	YARIS	PRÓŁ	KONRAD	ADAMCZYK	PAWEŁ	2004-12-30	2005-01-14	675
7	VECTRA	PRÓŁ	KONRAD	TWAROŻEK	PIOTR	2006-01-03	2006-01-02	80
8	S600	MYŚLIWIEC	WIESŁAW	TWAROŻEK	PIOTR	2006-01-02	2006-12-30	72400
9	PUNTO	NOWAK	PIOTR	CIEŚŁAK	JAN	2006-02-15	2007-01-01	8000
11	440	PRÓŁ	KONRAD	BRACKI	BOGDAN	2006-02-16	2006-12-01	4590
13	VECTRA	MYŚLIWIEC	WIESŁAW	CIEŚŁAK	JAN	2021-12-07	2021-12-31	230
14	PUNTO	NOWAK	PIOTR	CIEŚŁAK	JAN	2021-12-15	2022-12-13	2300
15	S600	MYŚLIWIEC	WIESŁAW	CIEŚŁAK	JAN	2022-01-03	2022-01-13	412
16	S600	NOWAK	PIOTR	NOWAK	KIESŁAW	2022-01-04	2022-01-13	341
17	YARIS	TUSK	NATALIA	KRÓŁ	MONIKA	2022-01-18	2022-01-26	300
18	PUNTO	BRANIECKI	ALBERT	STACUCH	PAWEŁ	2022-01-04	2022-01-08	350
19	VECTRA	BRANIECKI	ALBERT	KOWALSKI	RAFAŁ	2021-01-21	2022-01-11	640
20	S600	TUSK	NATALIA	NOWAK	KIESŁAW	2022-01-18	2022-01-19	40
22	Ka	PRÓŁ	KONRAD	KAPECKI	ADRIAN	2022-02-02	2022-02-26	350

Dodaj

Usuń

Szczegóły

☐ Edytowalne

☒ Constraint

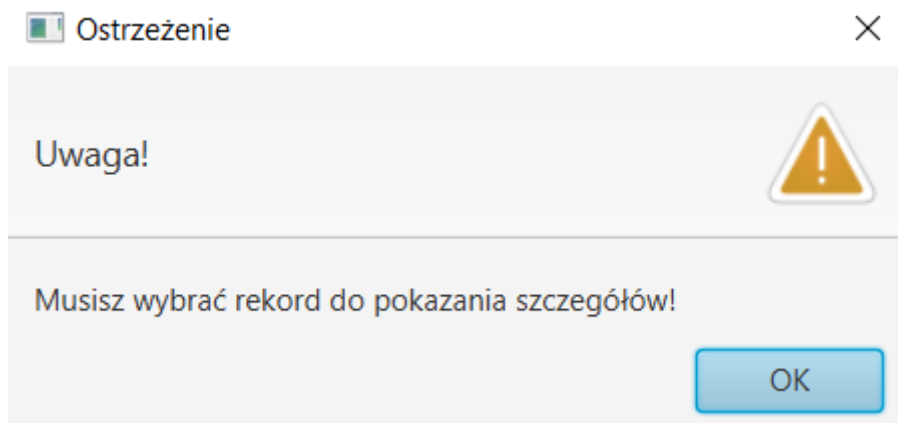
Szczegóły:

Przycisk „Szczegóły” pokazuje powiązane tabele dla danego rekordu.

Podgląd powiązanych tabel		— □ ×	
WYPOŻYCZENIA		SAMOCHODY	
ID_WYP: 11		ID_SAM: 160	
SAM: 440		NR_REJ: PKT-0967	
PRAC: PRÓL		MARKA: VOLVO	
KLIENT: BRACKI		MODEL: 440	
DATA_WYP: 2006-02-16		ROK_PROD: 1984	
DATA_ZWR: 2006-12-01		KRAJ_PROD: SZWECJA	
KOSZT: 4590		POJ_SIL: 1	
		KOSZT_DN: 30	
PRACOWNICY		KLIENCI	
ID_PRAC: 3		ID_KLI: 25	
NAZWISKO: PRÓL		NAZWISKO: BRACKI	
IMIE: KONRAD		IMIE: BOGDAN	
TELEFON: 942150521		NR_DOW: CC3478690	
PESEL: 21252196224		MIEJSC: GDANSK	
		ULICA: BALTYCKA 67/4	

Id wypożyczenia jest wpisywane setterem do pola statycznego, które później wykorzystywane jest w innym kontrolerze, żeby wykonać odpowiednie zapytanie do serwera MySQL.

Jeśli nie wybierzemy żadnego rekordu do pokazania szczegółów to pokaże nam się ostrzeżenie:

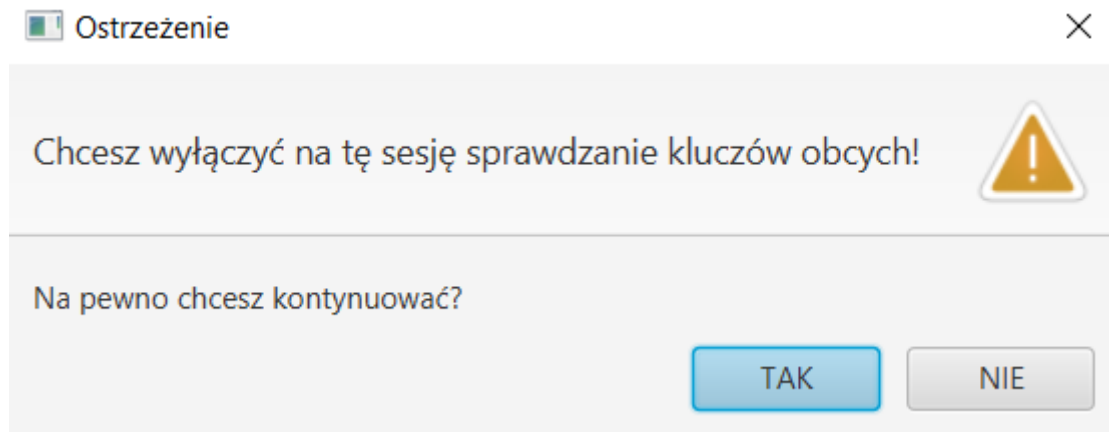


Constraint:

Funkcja przeznaczona do debugowania bądź testowania.

Wyłącza ona sprawdzanie kluczy obcych.

Jest to niebezpieczny proces, dlatego program pyta użytkownika, czy chce to na pewno zrobić.



5. Wycinki kodów:

Zapytanie do bazy na szczegóły:

```
ResultSet rs = stmt.executeQuery("SELECT wypozyczenia.id_wyp, samochody.model, pracownicy.nazwisko, klienci.nazwisko, wypozyczenia.data_wyp, wypozyczenia.data_zwr, "
+ " wypozyczenia.koszt, samochody.*, pracownicy.*, klienci.* FROM wypozyczenia, samochody, pracownicy, klienci WHERE wypozyczenia.id_sam = samochody.id_sam "
+ "AND wypozyczenia.id_prac = pracownicy.id_prac AND wypozyczenia.id_kli = klienci.id_kli AND `id_wyp` LIKE '"+zadania.DbAccess.getId()+"';");
```

Kod na comboboxa z klasą klientów

```
@FXML
private ComboBox<Kli> combobox3;

@FXML
ObservableList<Kli> oblist3 = FXCollections.observableArrayList();

public ObservableList<Kli> IDKLIFUNC() throws ClassNotFoundException, SQLException {
    Statement stmt = zadania.DbAccess.DbAccess();

    ResultSet rs = stmt.executeQuery("SELECT id_kli, nazwisko, imie FROM klienci");

    oblist3.clear();

    while(rs.next()) {
        oblist3.add(new Kli(rs.getInt("id_kli"), rs.getString("nazwisko"), rs.getString("imie")));
    }

    return oblist3;
}
```

Kod do usuwania rekordów

```
@FXML
void usuwanie(ActionEvent event) throws ClassNotFoundException, SQLException {

    Statement stmt = zadania.DbAccess.DbAccess();
    Wypozyczenia wypozyczenie = tableView.getSelectionModel().getSelectedItem();
    Klienci klient = tableView1.getSelectionModel().getSelectedItem();
    Samochody samochod = tableView2.getSelectionModel().getSelectedItem();
    Pracownicy pracownik = tableView3.getSelectionModel().getSelectedItem();

    if((WYPOZYCZENIA.isSelected() && tableView.getSelectionModel().isEmpty())
        || (KLIENCI.isSelected() && tableView1.getSelectionModel().isEmpty())
        || (SAMOCHODY.isSelected() && tableView2.getSelectionModel().isEmpty())
        || (PRACOWNICY.isSelected() && tableView3.getSelectionModel().isEmpty())) {

        Alert alert = new Alert(AlertType.WARNING, "Musisz najpierw wybrać rekord do usunięcia", ButtonType.OK);
        alert.setTitle("Ostrzeżenie");
        alert.setHeaderText("Nie wiadomo co usunąć!");
        ((Stage) alert.getDialogPane().getScene().getWindow()).setAlwaysOnTop(true);
        alert.showAndWait();
        return;
    }

    if (WYPOZYCZENIA.isSelected()) {
        int id = wypozyczenie.getId_wyp();
        stmt.executeUpdate("DELETE FROM `wypozyczenia` WHERE `id_wyp`= '" + id + "'");
        refresh();
    } else if (KLIENCI.isSelected()) {
        int id = klient.getId_kli();
        stmt.executeUpdate("DELETE FROM `klienci` WHERE `id_kli`= '" + id + "'");
        refresh2();
    } else if (SAMOCHODY.isSelected()) {
        int id = samochod.getId_sam();
        stmt.executeUpdate("DELETE FROM `samochody` WHERE `id_sam`= '" + id + "'");
        refresh3();
    } else if (PRACOWNICY.isSelected()) {
        int id = pracownik.getId_prac();
        stmt.executeUpdate("DELETE FROM `pracownicy` WHERE `id_prac`= '" + id + "'");
        refresh4();
    }
}
```

Kod odpowiadający za szukanie w tabelach

```
@FXML
void szukaj(ActionEvent event) throws ClassNotFoundException, SQLException {
    if(textfieldSZ.getText().trim().equals("")) { zadania.DbAccess.showAlertWarning(null, "Podaj frazę do wyszukania!"); return; }
    if (comboBoxSZ.getValue().equals("Wypożyczenia")) {
        if (comboBoxSZKOL.getSelectionModel().getSelectedIndex() == 3)
            refreshSZ("pracownicy.nazwisko", textfieldSZ.getText());
        else if (comboBoxSZKOL.getSelectionModel().getSelectedIndex() == 2)
            refreshSZ("klienci.nazwisko", textfieldSZ.getText());
        else
            refreshSZ(comboBoxSZKOL.getValue(), textfieldSZ.getText());
    } else if (comboBoxSZ.getValue().equals("Klienci")) {
        refreshSZ1(comboBoxSZKOL.getValue(), textfieldSZ.getText());
    } else if (comboBoxSZ.getValue().equals("Samochody")) {
        refreshSZ2(comboBoxSZKOL.getValue(), textfieldSZ.getText());
    } else if (comboBoxSZ.getValue().equals("Pracownicy")) {
        refreshSZ3(comboBoxSZKOL.getValue(), textfieldSZ.getText());
    }
}
```

Filtry do pól sprawdzające na bieżąco podane wartości

```
UnaryOperator<Change> telefonFilter = change -> {
    String newText = change.getControlNewText();
    if (newText.matches("^[0-9]{0,9}")) {
        return change;
    }
    return null;
};

UnaryOperator<Change> nrrejFilter = change -> {
    String newText = change.getControlNewText();
    if (newText.matches("^[A-Z-]{0,4}[A-Z0-9]{0,4}")) {
        return change;
    }
    return null;
};

UnaryOperator<Change> nrrowFilter = change -> {
    String newText = change.getControlNewText();
    if (newText.matches("^[A-Z]{0,2}[0-9]{0,7}+")) {
        return change;
    }
    return null;
};

UnaryOperator<Change> peselFilter = change -> {
    String newText = change.getControlNewText();
    if (newText.matches("^[0-9]{0,11}")) {
        return change;
    }
}
```

Kod odpowiedzialny za edycję id samochodu. Sprawdza on, czy wpisana nowa wartość jest pusta, czy powtarza się z jakimś innym id oraz czy podano prawidłową liczbę, a nie np. literę.

```
@FXML
void edit3sam(CellEditEvent<Samochody, Integer> event) throws ClassNotFoundException, SQLException {
    Statement stmt = zadania.DbAccess.DbAccess();
    Samochody samochod = tableView2.getSelectionModel().getSelectedItem();
    if(event.getNewValue() == null) { zadania.DbAccess.showAlertWarning(null, "Pole nie może być puste!"); refresh3(); return; }

    Boolean check = tableView2.getItems().stream().anyMatch(Check -> event.getNewValue().equals(Check.getId_sam()));
    if(event.getNewValue() == -1) { refresh3(); return; }

    if(!check)
        stmt.executeUpdate("UPDATE `samochody` SET `id_sam`='"+event.getNewValue()+"' WHERE id_sam = '"+samochod.getId_sam()+"'");
    else
        zadania.DbAccess.showAlertWarning(null, "Już jest takie id!");

    refresh3();
}
```

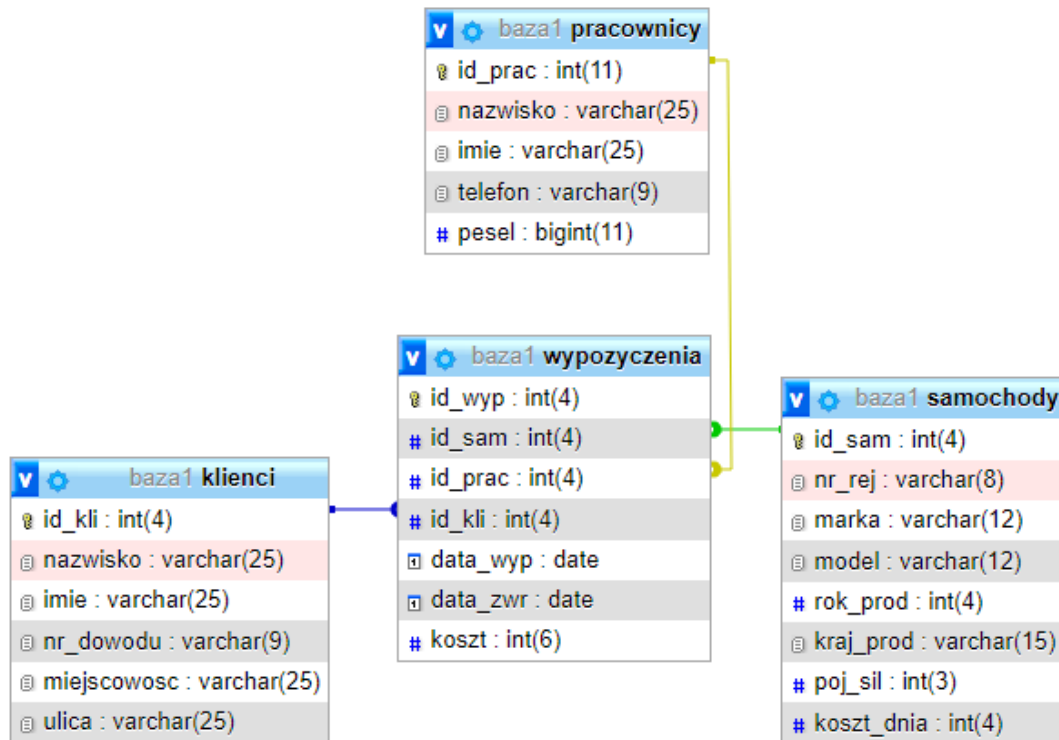
StringConverter użyty przy sprawdzaniu nieprawidłowego formatu danych (String zamiast Integera) przy edycji rekordu.

```
public class CustomIntegerStringConverter extends IntegerStringConverter {
    private final IntegerStringConverter converter = new IntegerStringConverter();
    Wypozyczenia wypozycczenie = tableView.getSelectionModel().getSelectedItem();

    @Override
    public String toString(Integer object) {
        try {
            return converter.toString(object);
        } catch (NumberFormatException e) {
            zadania.DbAccess.showAlertWarning(e, "Każde pole musi być uzupełnione poprawnie!");
        }
        return null;
    }

    @Override
    public Integer fromString(String string) {
        try {
            return converter.fromString(string);
        } catch (NumberFormatException e) {
            zadania.DbAccess.showAlertWarning(e, "Każde pole musi być uzupełnione poprawnie!");
            return -1;
        }
    }
}
```

6. Schemat ERD bazy danych



Moja baza danych składa się z czterech encji:

- wypożyczenia
- klienci
- samochody
- pracownicy

Tabela	Działanie	Rekordy	Typ	Metoda porównywania napisów	Rozmiar	Nadmiar
<input type="checkbox"/> klienci	Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń	11	InnoDB	utf8_polish_ci	16.0 KB	-
<input type="checkbox"/> pracownicy	Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń	6	InnoDB	utf8_polish_ci	32.0 KB	-
<input type="checkbox"/> samochody	Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń	11	InnoDB	utf8_polish_ci	16.0 KB	-
<input type="checkbox"/> wypozyczenia	Przeglądaj Struktura Szukaj Wstaw Opróżnij Usuń	16	InnoDB	utf8_polish_ci	64.0 KB	-
4 tabel	Suma	44	InnoDB	utf8_polish_ci	128.0 KB	0 B