**Kamil Filar**,

**Karol Bury**

*Informatyka II, lab 1*

23.05.2020r.

W CIĄGŁEJ AKTUALIZACJI

**Biblioteka**

***Bazy danych II***

**Oracle(PL/SQL), Java**

**Spis treści:**

Spis treści …………………………………………………………………………………………………….. 2

Diagram ERD …………………………………………………………………………………………………. 3

Opis i cel projektu …………………………………………………………………………………………. 4

Interfejs wraz z funkcjonalnościami ………………………………………………………………. 5

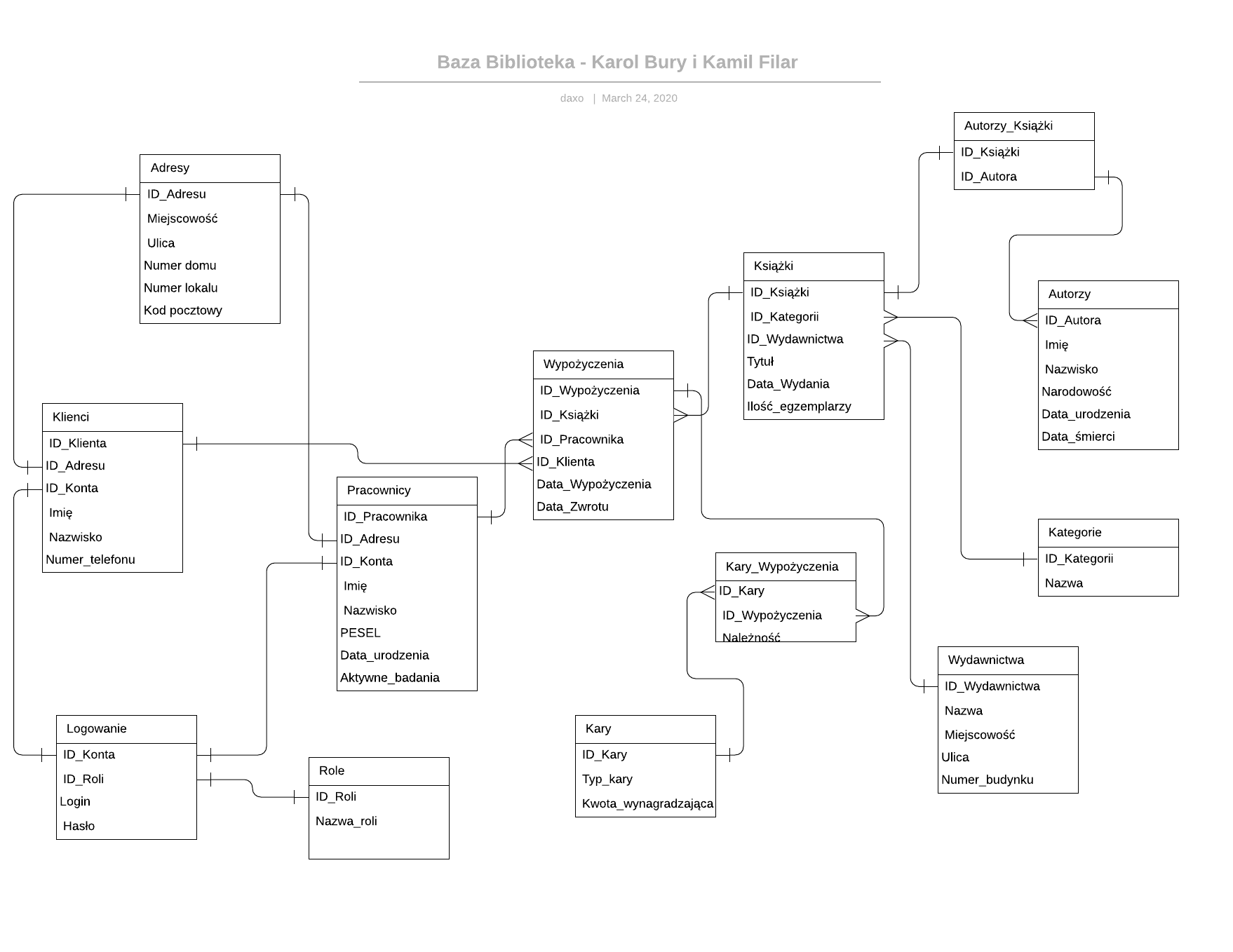
Logowanie …………………………………………………………………………………………… 5

Połączenie z bazą danych …………………………………………………………………….. 5

Wyświetlanie danych w obiektach jTable ……………………………………………. 6

Wymagania do korzystania z aplikacji …………………………………………………………… X

**Diagram ERD:**



**Opis i cel projektu:**

Projekt został stworzony w celu zarządzania i prowadzenia biblioteki. Program służy do administrowania bazy biblioteki przez pracownika. W projekcie został uwzględniony również interfejs dla klienta. Program jest oparty o system zarządzania relacyjną bazą danych Oracle, która zawiera trzynaście encji powiązanych ze sobą.

Aplikacja została stworzona w języku Java przy użyciu kompilatora NetBeans IDE 8.2 z wykorzystaniem technologii Java Swing oraz biblioteki ojdbc7.jar. Oferuje: połączenie z bazą danych, wyświetlanie autorów, wydawnictw, regulaminu oraz bazy książek, także panel rejestracyjny (dla klientów) oraz panel logowania. Panel administratora zawiera podgląd na całą bazę danych i umożliwia interakcje z nią (poprzez interakcję mamy na myśli dodawanie książek, aktualizowanie ich, a także autorów i wydawnictw). Dla klienta został przygotowany osobny interfejs, klient może dokonać rezerwacji książki, sprawdzić stan swoich wypożyczeń, a także przeglądnąć regulamin i jakie wydawnictwa, autorzy figurują w bazie biblioteki.

Baza danych została wykonana przy użyciu Oracle Database Express Edition (XE) Release 18.4.0.0.0 (18c) . Zawiera trzynaście encji oraz wykorzystuje: funkcje, procedury, trigery, kursory, rekordy, sekwencje oraz tablice.

**Interfejs aplikacji wraz z funkcjonalnościami:**

Interfejs dzieli się na XXX formatek, które służą do: logowania, wyświetlenia autorów, bazy książek, wydawnictw oraz regulaminu i listy wypożyczeń, sterowania GUI, rejestracji (która składa się z trzech formatek), a także formatka do edycji konta jak i panel administratora, który służy do zarządzania biblioteką.

**Połączenie z bazą danych:**

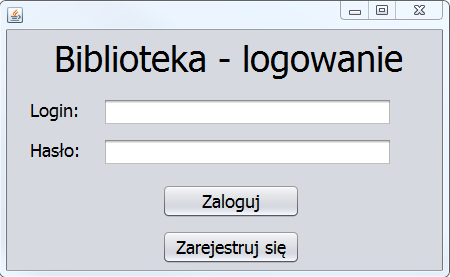
Do połączenia z bazą danych oracle wykorzystaliśmy klasę DbAccess.java, która wygląda następująco:



W formatkach wykorzystywaliśmy odwołanie do tej klasy, w celu łączenia się z bazą danych.

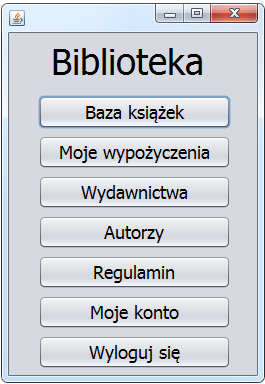
**Logowanie:**

Po uruchomieniu aplikacji pojawi się okno logowania, w którym musimy podać nasze dane w dwóch polach do tego przeznaczonych, w momencie podania błędnych danych aplikacja poinformuje nas zwracając odpowiedni komunikat. W procesie logowania, dzięki pobraniu ID\_Konta aplikacja odróżni czy logujemy się jako klient, czy jako administrator dzięki czemu zostanie wyświetlony odpowiedni panel startowy.



**Główny panel:**

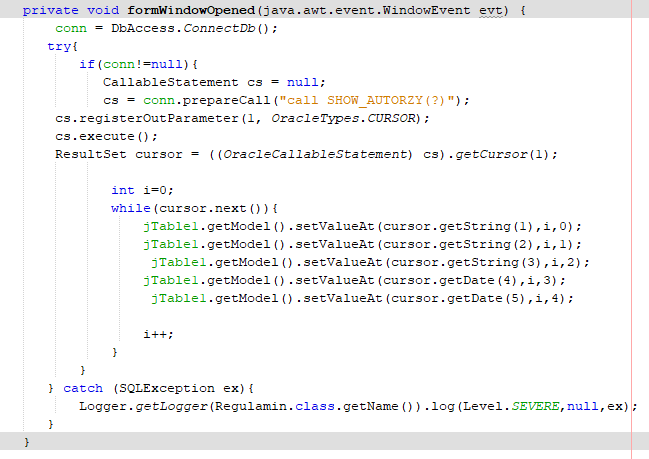
Po poprawnym zalogowaniu, zostajemy przekierowani do głównego panelu, z który służy do nawigacji po aplikacji.



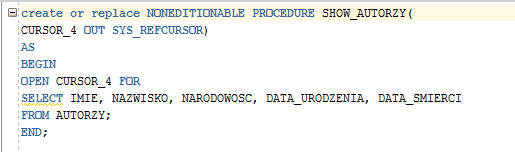
Widok dla klienta

**Wyświetlanie danych w obiektach jTable (na przykładzie formatki „Autorzy”):**

Do wyświetlenia danych w tabelkach użyliśmy następującej składni:



Dzięki zastosowaniu procedury SHOW\_AUTORZY za pomocą PL/SQL przekazaliśmy wszystkie wyniki do obiektu jTable. Podobnie wygląda klasa: Wydawnictwa.java i Regulamin.java (tam zostały zastosowane procedury SHOW\_WYDAWNICTWA oraz SHOW\_KARY).



Procedura SHOW\_AUTORZY.

**Baza książek:**

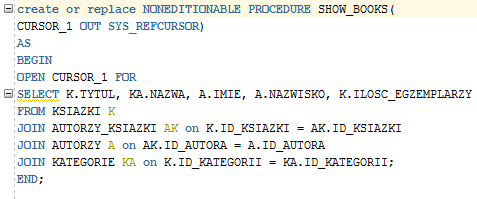
Jest to miejsce gdzie są wypisane wszystkie pozycje z biblioteki.

Tutaj użytkownik może wypożyczyć książkę po kliknięciu na wybraną przez siebie pozycję w tabeli (może ją również wyszukać w górnym TextFieldzie).

Do wypożyczenia użyliśmy funkcji ………………………

**ZDJ TEJ FUNKCJI**

Do wypisania ich użyliśmy procedury SHOW\_BOOKS, która wygląda następująco:



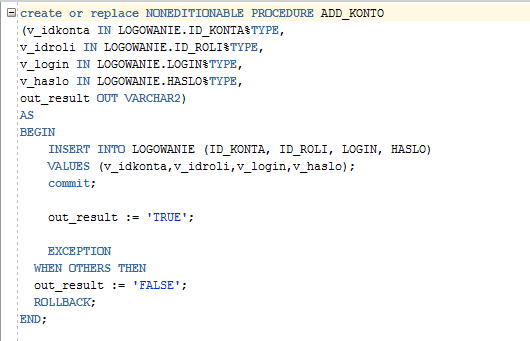
**Panel rejestracyjny:**

Plik Panel\_rejestracyjny.java

Zawiera pierwszy krok w celu utworzenia konta, w tym miejscu w aplikacji tworzymy nowe konto, gdzie użytkownik podaje swój login i hasło.

**ZDJ OKIENKA PANEL REJESTRACYJNY**

Do dodania danych do tabeli Logowanie użyliśmy funkcji ADD\_KONTO:

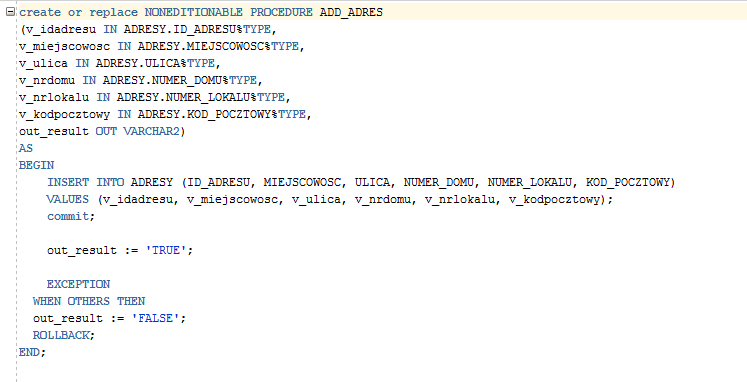


Plik Panel\_rejestracyjny2.java

Zawiera drugi krok w celu utworzenia konta, w tym miejscu w aplikacji podajemy adres użytkownika (miejscowość, ulica, numer domu, numer lokalu i kod pocztowy).

**ZDJ OKIENKA PANEL REJESTRACYJNY2**

Do dodania danych do tabeli Logowanie użyliśmy funkcji ADD\_ADRES:

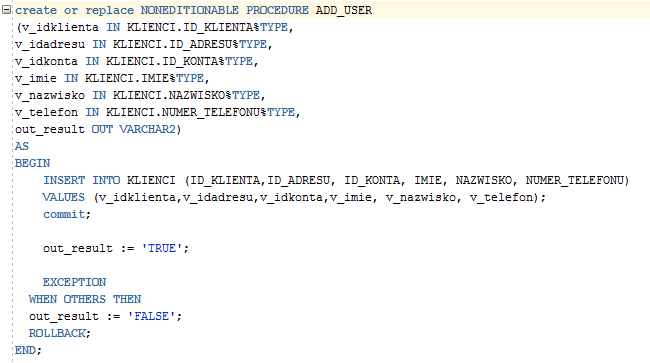


Plik Panel\_rejestracyjny3.java

Zawiera drugi trzeci, ostatni krok w celu utworzenia konta, w tym miejscu w aplikacji podajemy dane użytkownika (imię, nazwisko oraz numer telefonu).

**ZDJ OKIENKA PANEL REJESTRACYJNY3**

Do dodania danych do tabeli Logowanie użyliśmy funkcji ADD\_USER:



Do automatycznej inkrementacji numeru ID konta, klienta i adresu użyliśmy sekwencji IDKONTA, IDKKLIENTA, IDADRESU:

https://media.discordapp.net/attachments/689800346004095003/712938134144286801/unknown.png

**Wymagania do korzystania z aplikacji:**

**Dane logowania** (konto administratora):

Login: **admin**

Hasło: **admin**

**Dane logowania** (przykładowe konto użytkownika):

Login: **student**

Hasło: **student**

**Załączniki:**

Folder **docs:**

* **Diagram ERD.png** (zawiera schemat ERD bazy danych)
* **Funkcjonalność\_bazy\_biblioteka.docx** (zawiera opis funkcjonalnośći)

Folder **gui:**

* Biblioteka (folder zawiera aplikację)

Folder **pass**:

* **pass.txt** (zawiera hasła do kont używanych w aplikacji)

Folder **sql:**

* **biblioteka.sql** (plik z bazą danych)
* **procedures.txt** (plik z procedurami)
* **sequences.txt** (plik z sekwencjami)