Krzysztof Ćmiel

*Informatyka II, lab 1*

06.06.2018r.

**Wypożyczalnia Samochodowa**

Bazy danych II

Oracle(pl/sql)

Java

Spis treści

[Spis treści 2](#_Toc516063710)

[Diagram ERD 3](#_Toc516063711)

[Opis i cel projektu 4](#_Toc516063712)

[Interfejs aplikacji wraz z funkcjonalnościami 5](#_Toc516063713)

[Logowanie 5](#_Toc516063714)

[Klienci 6](#_Toc516063715)

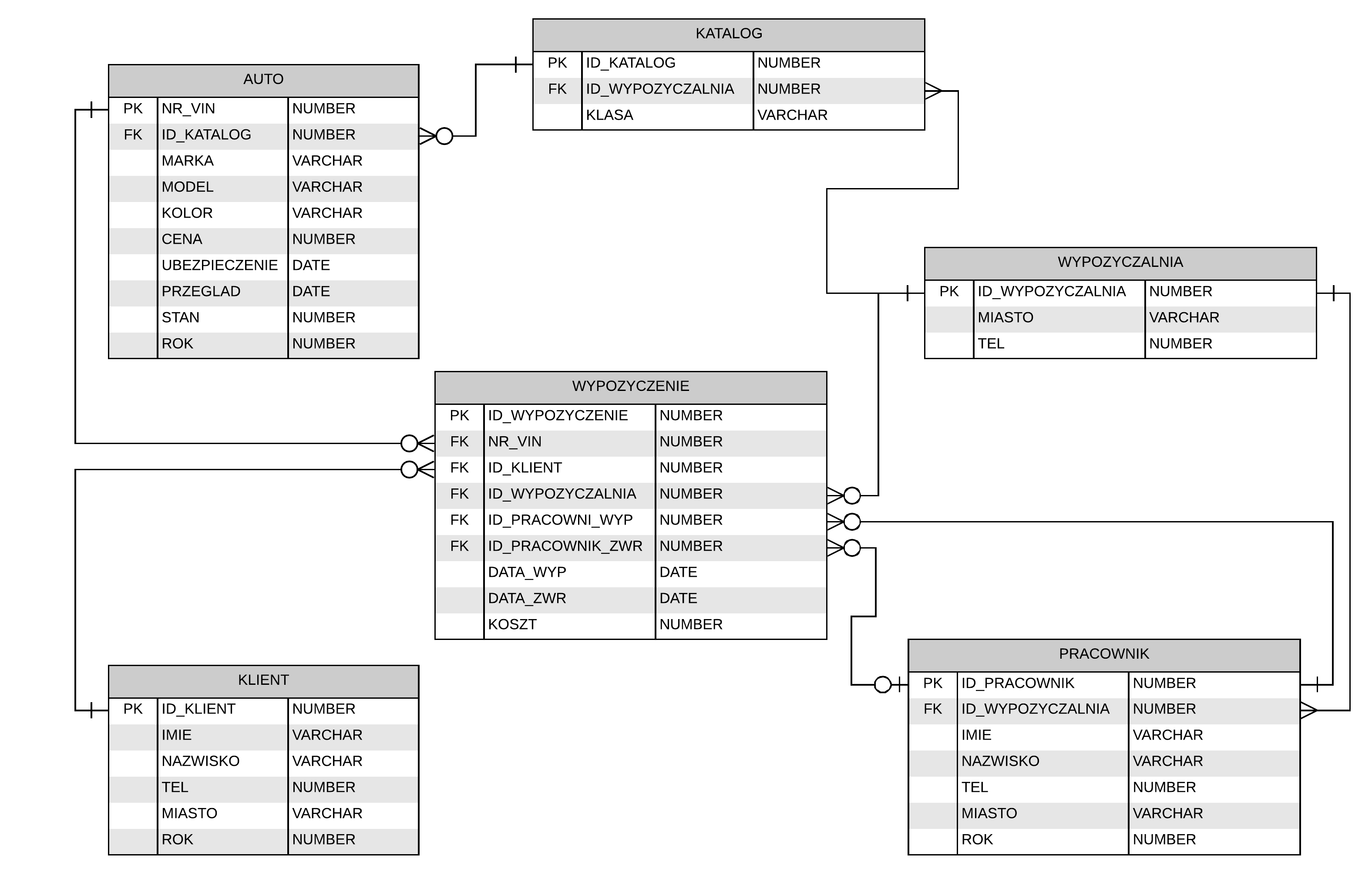
[Katalog aut 7](#_Toc516063716)

[Wypożyczenia trwające 8](#_Toc516063717)

[Wypożyczenia zakończone 10](#_Toc516063718)

[Wymagania do korzystania z aplikacji 12](#_Toc516063719)

# Diagram ERD



# Opis i cel projektu

Projekt został stworzony w celu zarządzania wypożyczalnią samochodową z trzema placówkami (Rzeszów, Kraków oraz Warszawa) przez pracowników. Jest oparty o system zarządzania relacyjną bazą danych Oracle, która zawiera 6 encji powiązanych ze sobą.

Aplikacja została stworzona w języku Java. Oferuje: połączenie z bazą danych, wyświetlanie (klientów, aut, wypożyczeni trwających oraz zakończonych), dodawanie klientów, obliczanie przykładowej oraz ostatecznej ceny wypożyczenia, dodawanie nowego wypożyczenia i zakończenie trwającego.

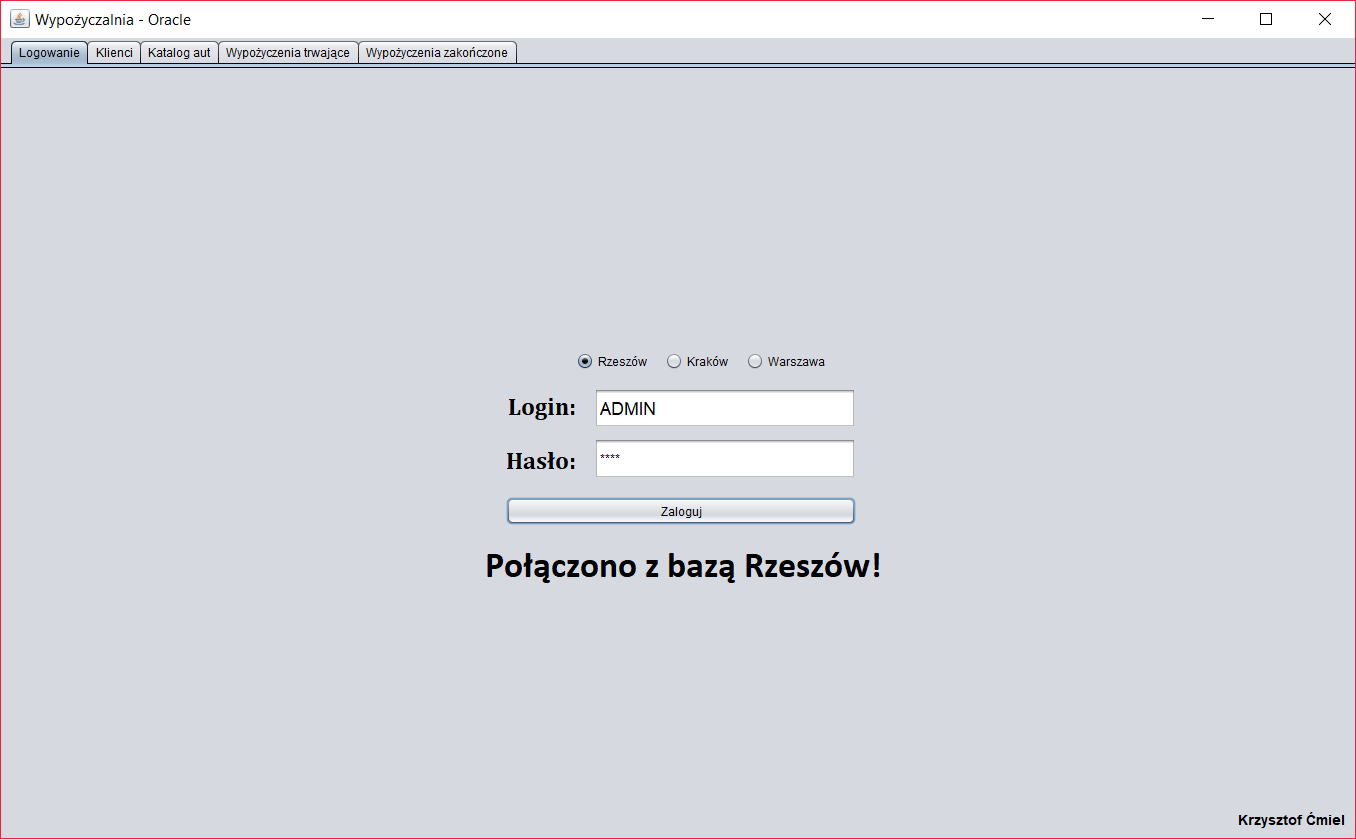
Baza danych wykonana została przy użyciu Oracle Database 11g XE. Zawiera 6 encji oraz wykorzystuje: funkcje, procedury, pakiety, sekwencje, trigery, kursory, rekordy oraz tablice.

# Interfejs aplikacji wraz z funkcjonalnościami

Aplikacja została zaprojektowana w technologii Java Swing przy pomocy środowiska NetBeans IDE 8.2, bibliotek oracle-jdbc oraz rs2xml.

Interfejs dzieli się na 5 zakładek: logowanie, klienci, katalog aut, wypożyczenia trwające, wypożyczenia zakończone.

## Logowanie

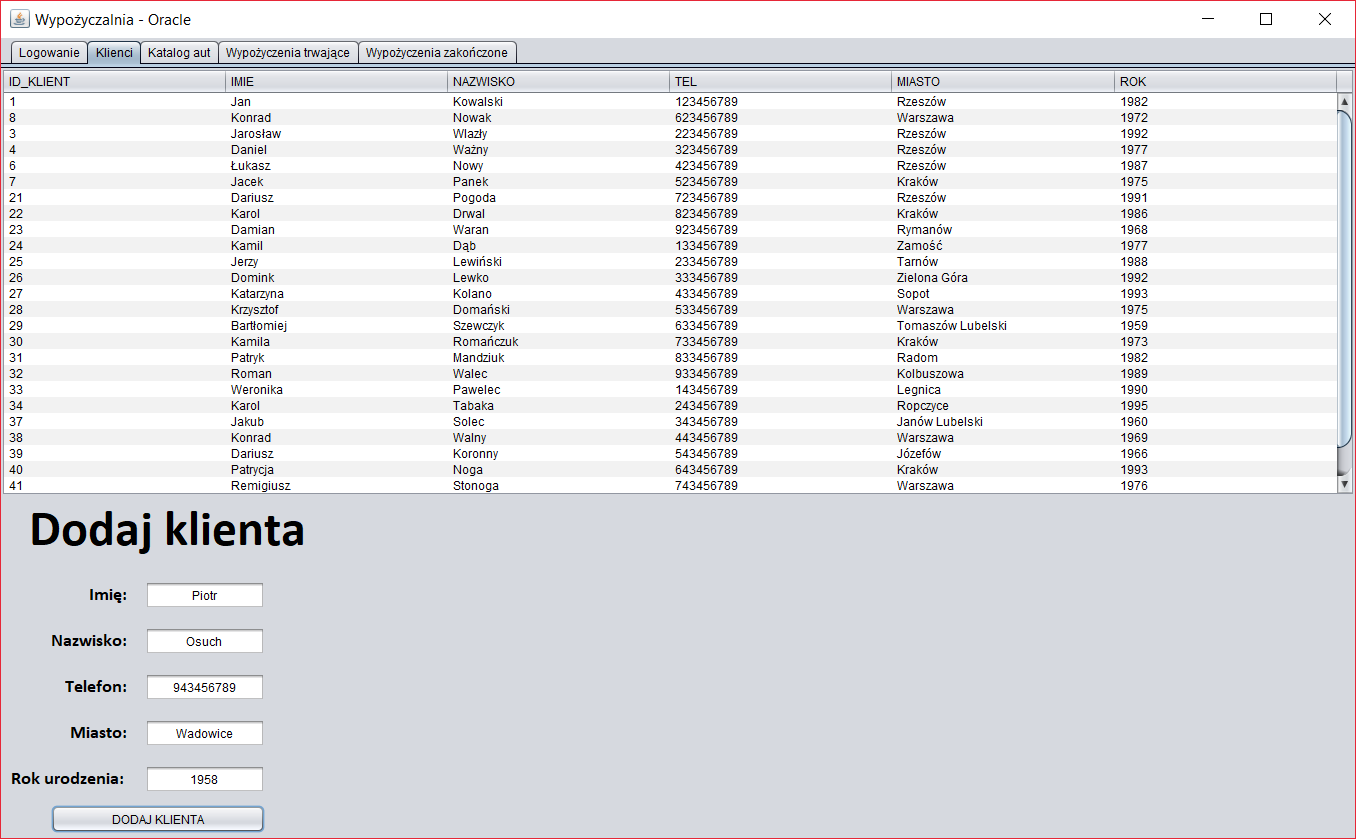


Po uruchomieniu aplikacji pojawi się zakładka logowania. Mamy do wyboru trzy oddziały wypożyczalni. Jeśli nie wybierzemy żadnej z nich, aplikacja przypomni nam o tym wyświetlając odpowiedni komunikat. Są dwa pola tekstowe gdzie podajemy login oraz hasło. Jeśli podczas łączenia wystąpi błąd, wyskoczy okno z informacją o problemie.

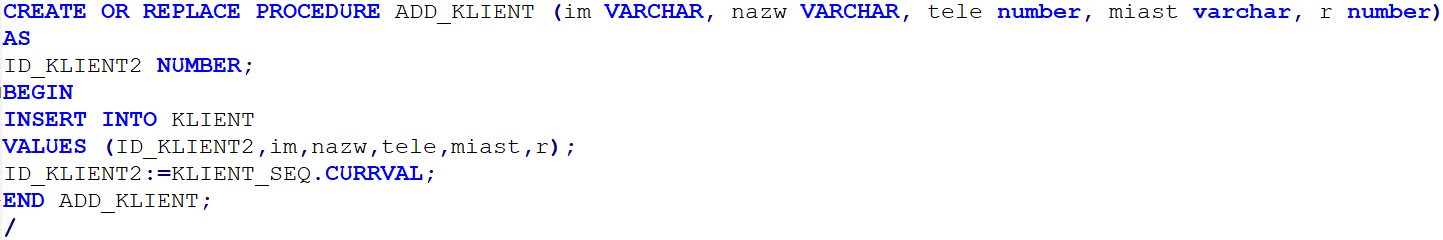
Kod jednej z metod, które łączą się bazą danych oraz uzupełniają tabele w pozostałych zakładkach:



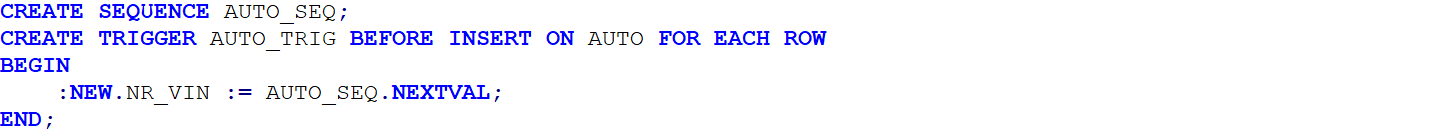
## Klienci



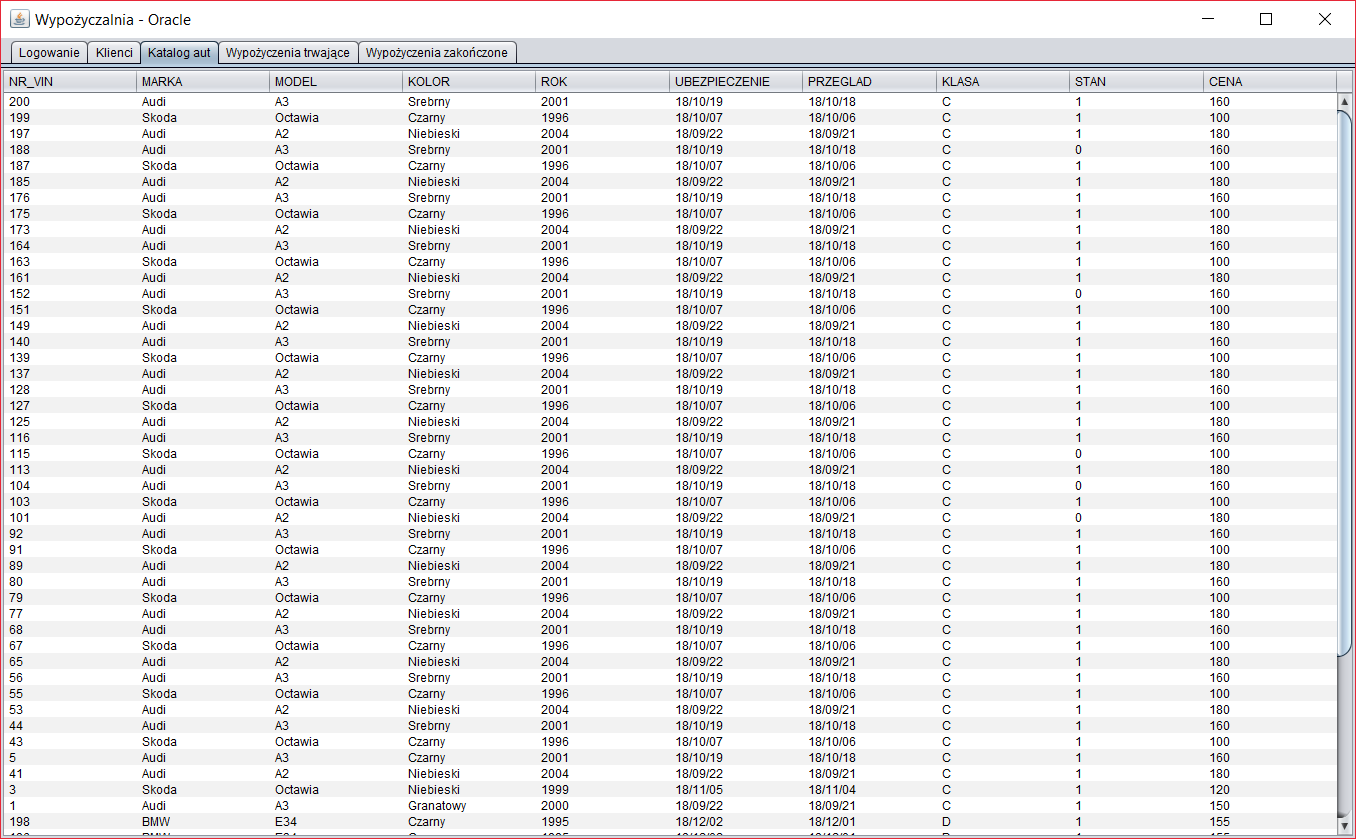
W zakładce Klienci za pomocą tabeli wyświetlane są informacje o klientach ze wszystkich oddziałów wypożyczalni. Mamy możliwość również dodania nowego klienta. W tym celu wykorzystana jest procedura o nazwie *ADD\_KLIENT*, która pobiera wartości wprowadzone w aplikacji i dodaje nowy wiersz w encji *KLIENT*.



Zamiast wprowadzać ręcznie id klienta, wykorzystana została sekwencja wraz z trigerem, która automatycznie inkrementuje pole id podczas dodawania.

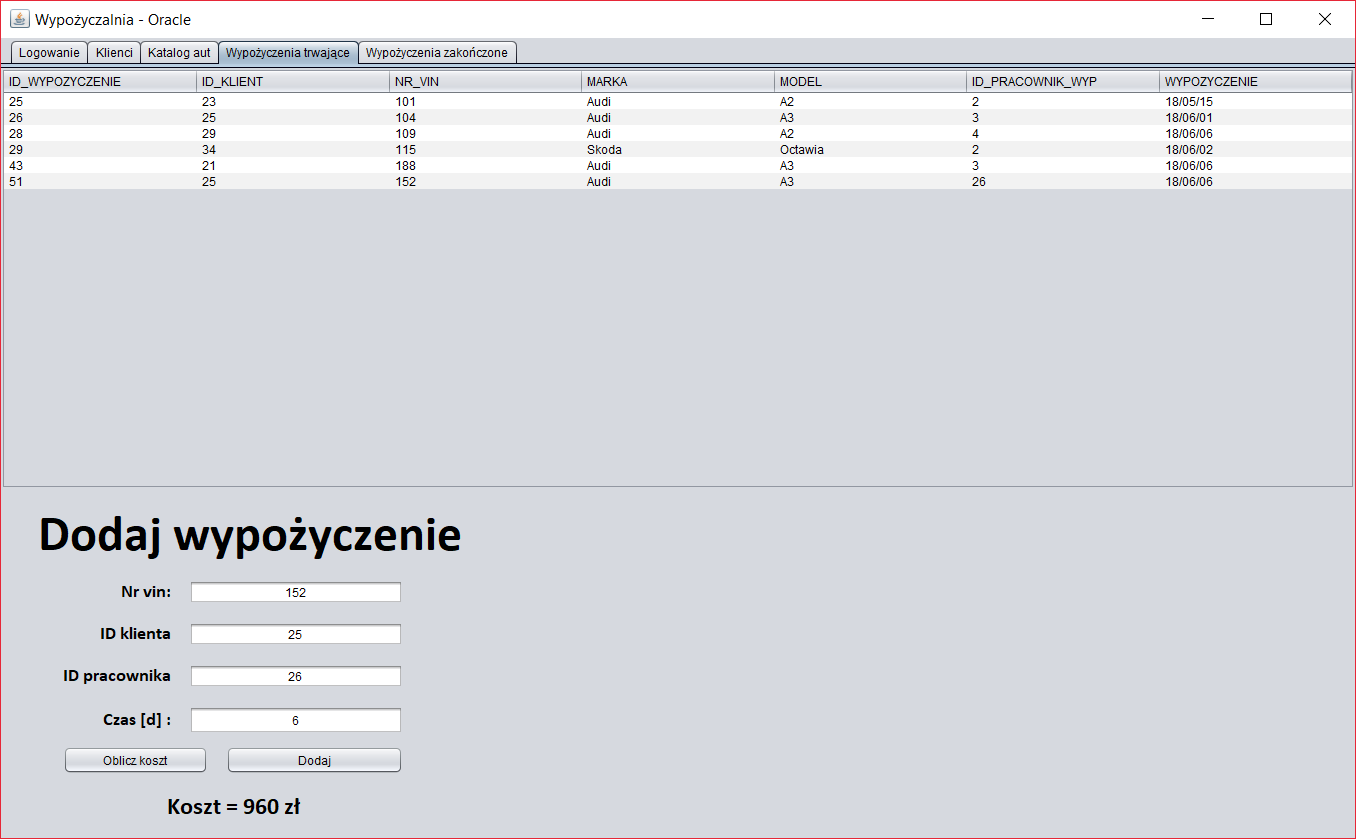


## Katalog aut



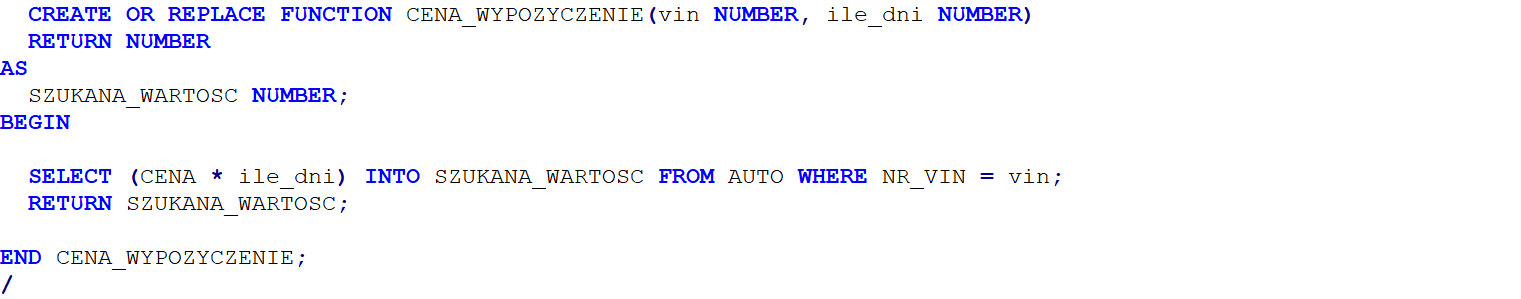
W tej zakładce wyświetlane są wszystkie auta przypisane do wypożyczalni, którą wybraliśmy podczas logowania.

## Wypożyczenia trwające



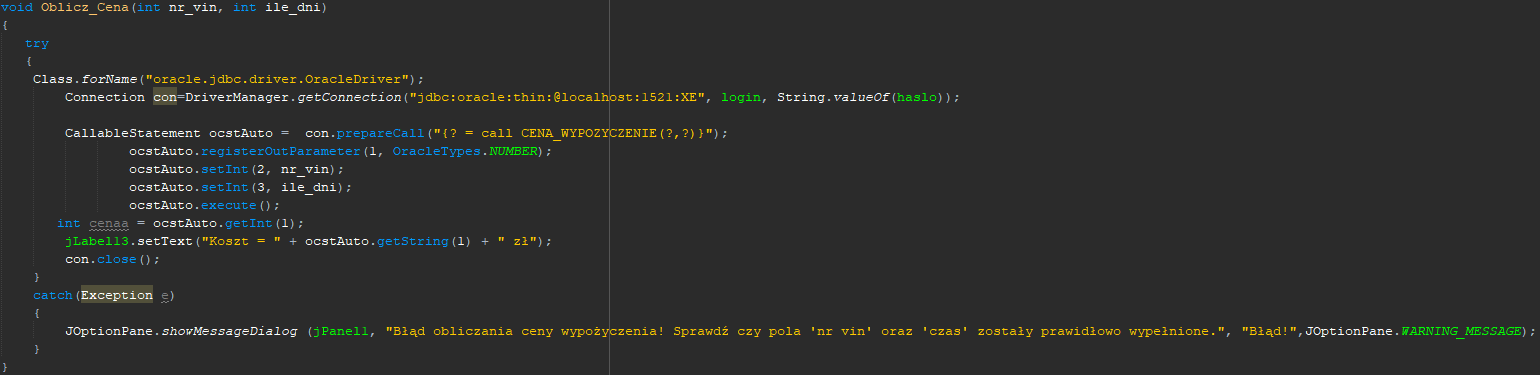
W zakładce Wypożyczenia trwające wypisane są w tabeli wypożyczenia, które jeszcze nie zostały zakończone. Podczas dodawania, wypożyczenia, po podaniu planowanego czasu wypożyczenia możemy sprawdzić jaki będzie jego koszt.

W tym celu została wykorzystana funkcja *CENA\_WYPOZYCZENIE*:

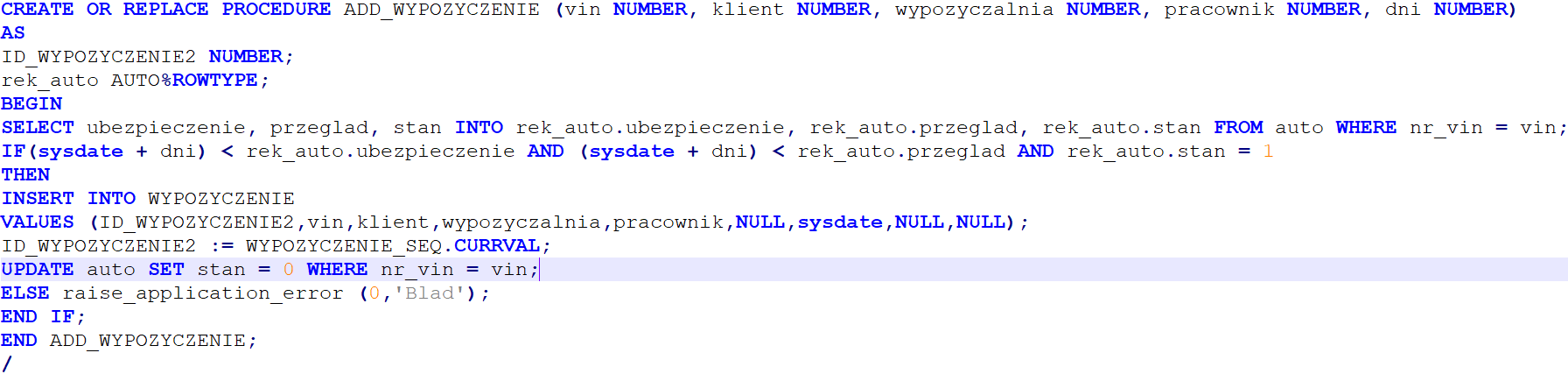


Pobiera ona z aplikacji nr vin oraz liczbę planowanych dni. Następnie mnoży cenę auta o takim numerze vin przez jego cenę i zwraca wynik tego działania. Wynik obliczenia wypisywany jest w aplikacji.

Wymiana wartości pomiędzy aplikacją a funkcją w bazie danych była możliwa dzięki metodzie *Oblicz\_Cena:*

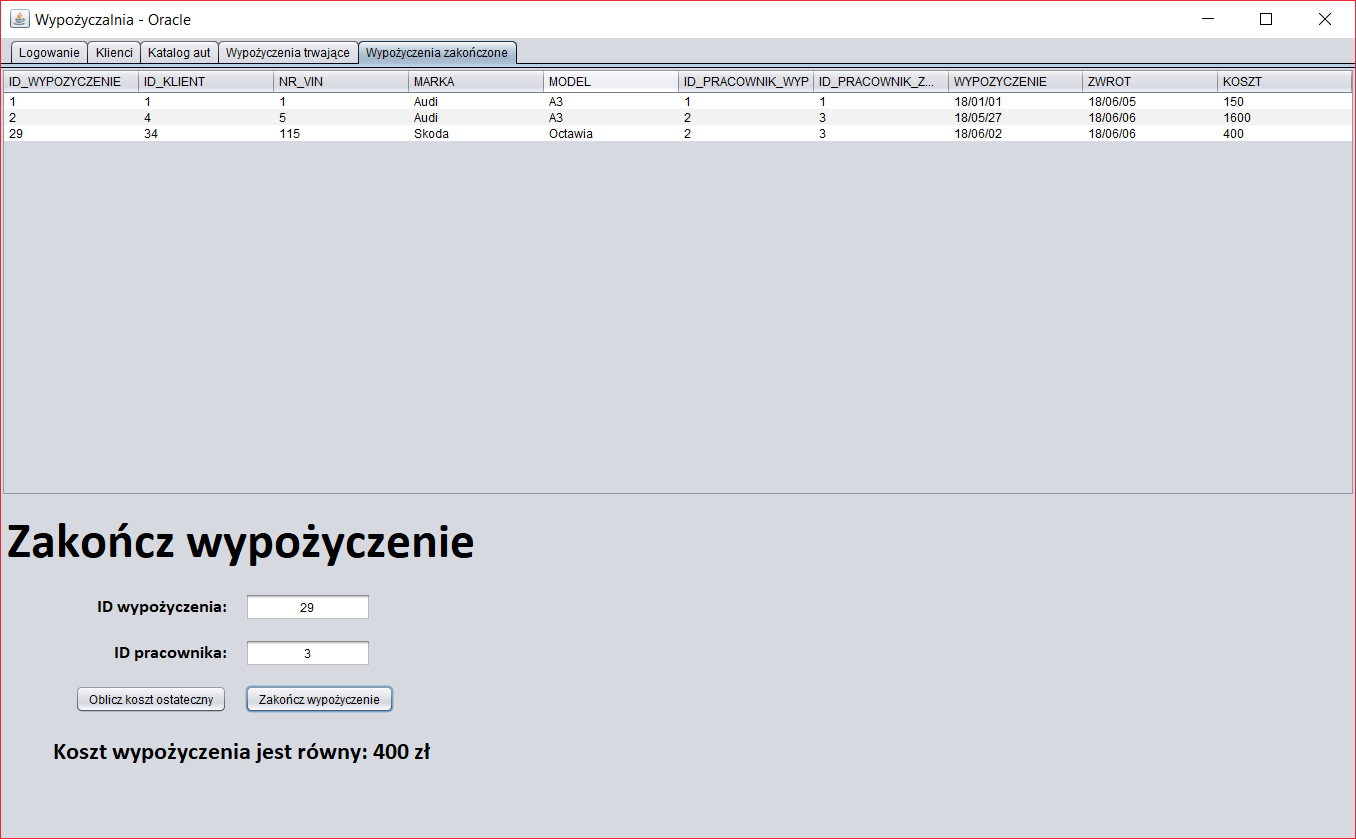


Do samego dodawania wypożyczenia została użyta procedura *ADD\_WYPOZYCZENIE:*

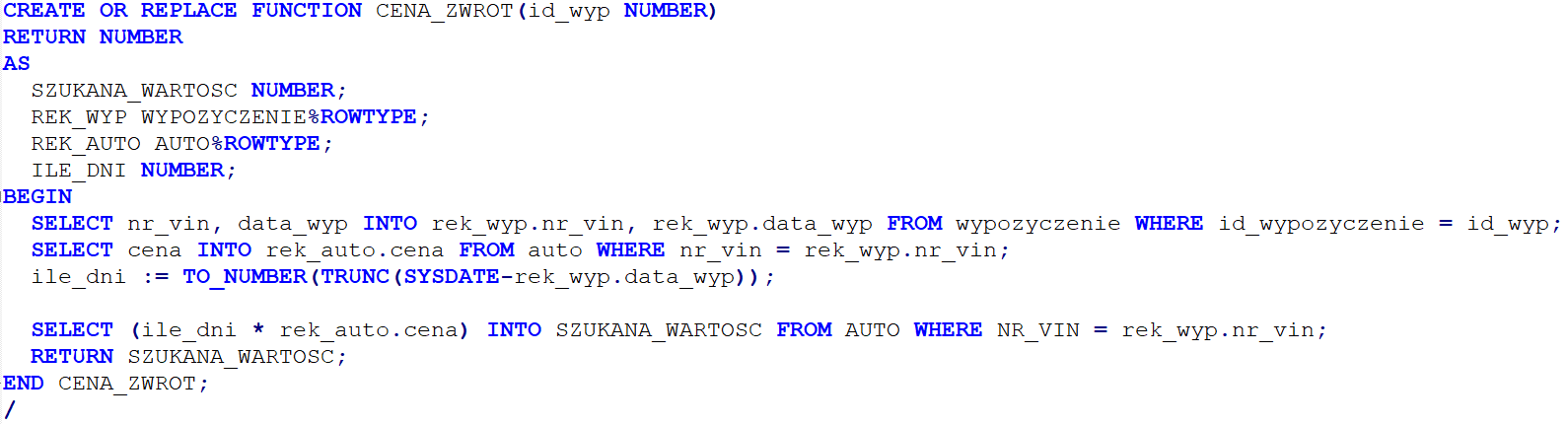


Procedura ta pobiera z aplikacji nr vin, id klienta, id wypożyczalni, id\_pracownika oraz liczbę planowanych dni wypożyczenia. Za pomocą rekordu wczytuje wiersz z encji *AUTO.* Następnie sprawdza czy w planowanym okresie wypożyczenia będzie aktualne ubezpieczenie oraz badania techniczne. Id wypożyczenia jest wpisywane automatycznie dzięki sekwencji wraz z trigerem. Na końcu procedury stan wypożyczanego auta jest zmieniany na 0. Jeśli nie będą spełnione warunki związane z ubezpieczeniem oraz badaniami technicznymi, na interfejsie aplikacji ukaże się okno informujące o problemie.

## Wypożyczenia zakończone



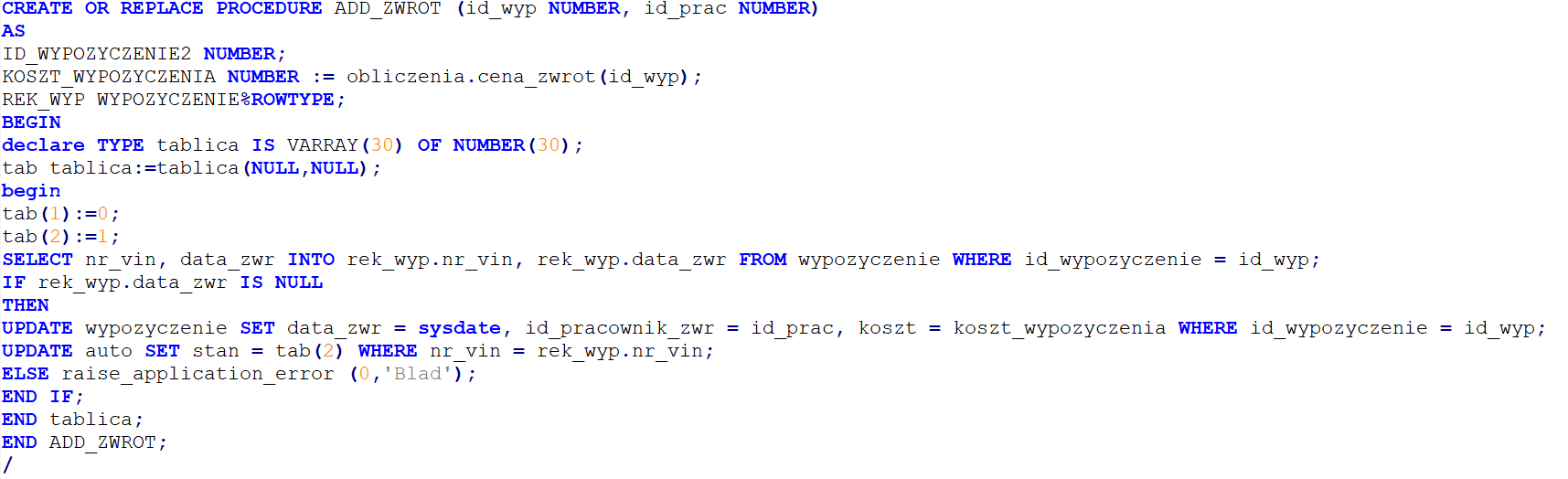
W tej zakładce wyświetlane są wypożyczenia zakończone w oddziale, który wybraliśmy podczas logowania. Przed zakończeniem wypożyczenia mamy możliwość sprawdzenia obecnego kosztu wypożyczenia dzięki funkcji *CENA\_ZWROT:*



Funkcja ta pobiera id wypożyczenia, pobiera dwa rekordy z encjii *WYPOZYCZENIE*  oraz *AUTO. Następnie od aktualnej daty systemowej odejmowana jest data początkowa wypożyczenia.* Dzięki funkcji *TRUNC* usuwana jest część dziesiętna, a następnie za pomocą funkcji *TO\_NUMBER* wynik operacji jest przekonwertowany do typu *number*.

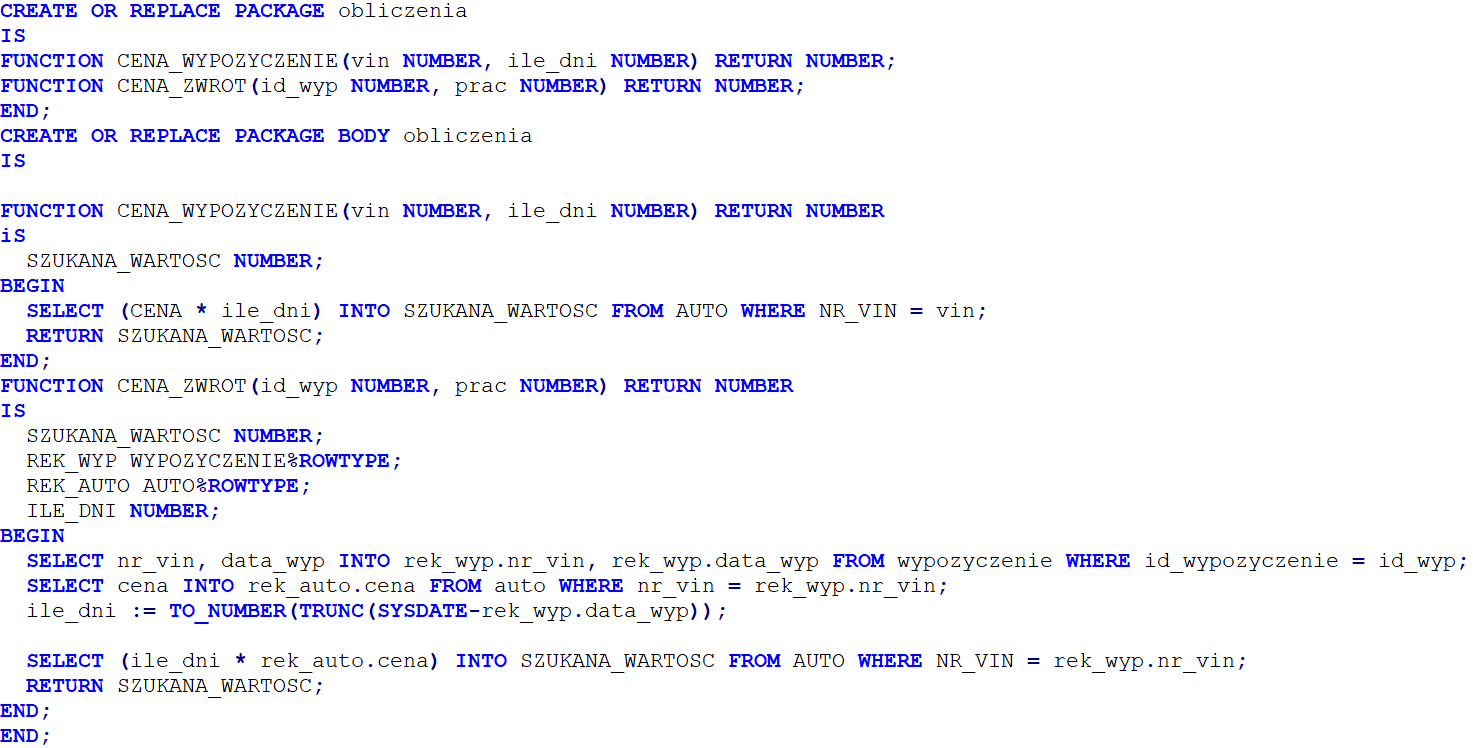
W celu zakończenia wypożyczenia podajemy id wypożyczenia oraz id pracownika.

Aby było możliwe zakończenie wypożyczenia, została wykorzystana procedura *ADD\_ZWROT*



Funkcja ta pobiera z aplikacji id wypożyczenia oraz id pracownika. Następnie przypisuje do zmiennej wynik funkcji *CENA\_ZWROT* z parametrem id wypożyczenia, która jest zawarta w pakiecie *OBLICZENIA*. Kolejno wczytuje rekord z encji *WYPOZYCZENIE.* W funkcji wykorzystana jest również tablica która dla dwóch swoich indeksów przyjmuje wartość 0 i 1. Wartości wczytane z tej tablicy są wykorzystywane podczas zmiany stanu auta pod koniec procedury. Procedura przed aktualizacją wypożyczenia sprawdza czy wypożyczenie o podanym id nie jest już zakończone. Jeśli nie, funkcja zwraca błąd i na interfejsie aplikacji wyświetla się okno z informacją o błędzie.

Pakiet *OBLICZENIA:*



# Wymagania do korzystania z aplikacji

Dane logowania:

Login: **ADMIN**

Hasło: **pass**

# Załączniki

* Function.txt (kod funkcji)
* Procedure.txt (kod procedur)
* Sequence.txt (kod sekwencji)
* Package.txt (kod pakietów)
* Constraint.txt (kod tworzenia kluczy obcych)
* Create.txt (kod tworzenia encji)
* Diagram\_erd.png
* Folder „Wypozyczalnia1” (projekt aplikacji)