uniwersytet łódzki, wydział ekonomiczno-socjologiczny, informatyka ekonomiczna rok ii semestr iv, rok akademicki 2021/2022

PROWADZĄCY: dr inż. sebastian łacheciński

Projekt bazy danych na przedmiot Programowanie Serwera Baz Danych

Nazwa bazy : RENTALEX

autorzy

Igor Owczarek, hubert warchoł

2022

Spis treści

[Opis tła projektowanego i implementowanego zagadnienia 2](#_Toc107946498)

[Opis wymagań funkcjonalnych i niefunkcjonalnych 3](#_Toc107946499)

[Wymagania funkcjonalne: 3](#_Toc107946500)

[Wymagania niefunkcjonalne: 4](#_Toc107946501)

[Schemat bazy danych w postaci graficznej 5](#_Toc107946502)

[Skrypt tworzenia bazy danych 6](#_Toc107946503)

[Wykaz utworzonych tabel fizycznych 7](#_Toc107946504)

[Skrypt tworzenia tabel i referencji między nimi 8](#_Toc107946505)

[Wykaz zaimplementowanych indeksów 13](#_Toc107946506)

[Skrypt tworzenia indeksów 14](#_Toc107946507)

[Skrypt dodający przykładowe dane do tabel 16](#_Toc107946508)

[Wykaz utworzonych sekwencji 27](#_Toc107946509)

[Skrypt tworzący sekwencję 28](#_Toc107946510)

[Wykaz utworzonych widoków 29](#_Toc107946511)

[Skrypt tworzenia widoków 30](#_Toc107946512)

[Wykaz utworzonych procedur składowanych 33](#_Toc107946513)

[Skrypt tworzenia procedur składowanych 34](#_Toc107946514)

[Wykaz utworzonych funkcji użytkownika 58](#_Toc107946515)

[Skrypt tworzenia funkcji definiowanych przez użytkownika 59](#_Toc107946516)

[Wykaz utworzonych triggerów 64](#_Toc107946517)

[Skrypt z tworzenia triggerów 64](#_Toc107946518)

[Wykaz obiektów wartości domyślnych oraz reguł 67](#_Toc107946519)

[Wykaz obiektów reguł 67](#_Toc107946520)

[Wnioski i podsumowanie 68](#_Toc107946521)

# Opis tła projektowanego i implementowanego zagadnienia

Wykonany przez nas projekt dotyczy aspirującej wypożyczalni samochodów „Rentalex”. W skład wypożyczalni wchodzą wszelkiego rodzaju samochody. Począwszy od samochodów dostawczych do super nowoczesnych samochodów sportowych. Wypożyczalnia oferuje możliwość rezerwacji samochodu z wyprzedzeniem. Aby obniżyć koszta, firma „Rentalex” prowadzi własny serwis dla swojej floty.

Baza danych przechowuje wszelkie informacje na temat wszystkich samochodów, które w ofercie posiada firma „Rentalex”. Zaimplementowano możliwość szybkiego dodawania nowych samochodów do bazy danych. Aby ułatwić identyfikacje, w bazie znajdują się nazwy marek i modeli najpopularniejszych aut, informacje na temat rodzajów nadwozia oraz na temat rodzaju silników. Klient ma możliwość wypożyczyć samochód w trzech różnych opcjach, w zależności od standardu wypożyczanego samochodu. Dostępne są trzy opcje tj. standard, business i premium. Każdy standard posiada inną cenę za wypożyczenie.

Zaimplementowana została możliwość składania rezerwacji. Rezerwacja może zostać zgłoszona z wyprzedzeniem oraz istnieje możliwość przypisania wielu rezerwacji do wielu klientów. Dodana została również funkcjonalność związana z wypożyczeniem samochodu.

Stworzona baza ułatwia ogólną obsługę samochodów, dzięki zaimplementowanym funkcjonalnościom tj. informacje na temat ubezpieczenia samochodu, informacja na temat przeglądu oraz na temat napraw.

Wszystkie wyżej wymienione elementy składają się na działającą bazę danych wypożyczalni samochodów „Rentalex”.

# Opis wymagań funkcjonalnych i niefunkcjonalnych

## Wymagania funkcjonalne:

* Dodawanie rezerwacji
* Dodawanie wypożyczenia
* Dodawanie faktury
* Modyfikacja faktury
* Usunięcie faktury
* Dodawanie klientów
* Modyfikacja klienta
* Dodawanie samochodu
* Modyfikacja samochodu
* Usunięcie samochodu
* Aktualizacja cen za wypożyczenie
* Dodanie nowego przeglądu
* Dodawanie informacji o naprawach
* Dodawanie ubezpieczenia
* Modyfikacja ubezpieczenia
* Usunięcie ubezpieczenia
* Dodanie płatności

## Wymagania niefunkcjonalne:

* Brak możliwości usunięcia klienta
* Brak możliwości usunięcia ubezpieczenia z bieżącego i poprzedniego roku
* Nie ma możliwości usunięcia faktury z bieżącego roku
* Nie można wypożyczyć samochodu bez przeglądu
* Nie można usunąć płatności z bieżącego roku

# Schemat bazy danych w postaci graficznej



# Skrypt tworzenia bazy danych

--skrypt tworzšcy bazę danych

USE master

GO

CREATE DATABASE rentalex

ON

PRIMARY

( NAME = wypozyczalnia\_samochodow\_dat1,

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\rentalex1.mdf',

SIZE = 100MB,

MAXSIZE = 1000MB,

FILEGROWTH = 10MB

),

( NAME = wypozyczalnia\_samochodow\_dat2,

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\rentalex2.mdf',

SIZE = 100MB,

MAXSIZE = 1000MB,

FILEGROWTH = 10MB

)

LOG ON

( NAME = wypozyczalnia\_samochodow\_log1,

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\rentalexlog1\_ldf',

SIZE = 50MB,

MAXSIZE = 100MB,

FILEGROWTH = 10%

);

# Wykaz utworzonych tabel fizycznych

Obraz zawierający stół

Opis wygenerowany automatycznie

# Skrypt tworzenia tabel i referencji między nimi

CREATE SCHEMA dict;

GO

CREATE TABLE dict.tbl\_wojewodztwo

(

IdWojewodztwa tinyint NOT NULL PRIMARY KEY,

Wojewodztwo varchar(25) NOT NULL

);

CREATE TABLE dict.tbl\_plec

(

IdPlci tinyint NOT NULL PRIMARY KEY,

Plec varchar(15) NOT NULL,

SymbolPlci char(1) NOT NULL

);

CREATE TABLE dict.tbl\_kraj

(

IdKraju int NOT NULL PRIMARY KEY,

Kraj varchar(50)

);

CREATE TABLE dict.tbl\_silnik

(

IdSilnika tinyint NOT NULL PRIMARY KEY,

TypSilnika varchar(50) NOT NULL,

);

CREATE TABLE dict.tbl\_rodzaj\_nadwozia

(

IdRodzajuNadwozia tinyint NOT NULL PRIMARY KEY,

RodzajNadwozia varchar(25) NOT NULL

);

CREATE TABLE dict.tbl\_marka

(

IdMarki int NOT NULL PRIMARY KEY,

Marka varchar(50)

);

CREATE TABLE dict.tbl\_model

(

IdModelu int NOT NULL PRIMARY KEY,

Model varchar(50)

);

CREATE TABLE dict.tbl\_rodzaj\_samochodu

(

IdRodzajuSamochodu tinyint NOT NULL PRIMARY KEY,

RodzajSamochodu varchar(25) NOT NULL,

Cena money

);

CREATE TABLE dict.tbl\_forma\_faktury

(

IdFormyFaktury tinyint NOT NULL PRIMARY KEY,

FormaFaktury varchar(15)

);

CREATE TABLE dict.tbl\_forma\_platnosci

(

IdFormyPlatnosci tinyint NOT NULL PRIMARY KEY,

FormaPlatnosci varchar(25)

);

--tabela klient

CREATE TABLE tbl\_klient

(

IdKlienta int NOT NULL PRIMARY KEY,

Imie varchar(30) NOT NULL,

Nazwisko varchar(50) NOT NULL,

Adres varchar(60) NOT NULL,

Miejscowosc varchar(60) NOT NULL,

KodPocztowy varchar(6) NOT NULL,

IdWojewodztwa tinyint NOT NULL,

IdKraju int NOT NULL,

Telefon varchar(15) NOT NULL,

Email varchar(40),

DataUrodzenia date NOT NULL,

IdPlci tinyint NOT NULL,

KodKlienta char(8) NULL

CONSTRAINT FK\_Wojewodztwo FOREIGN KEY (IdWojewodztwa) REFERENCES dict.tbl\_wojewodztwo(IdWojewodztwa),

CONSTRAINT FK\_klient\_plec FOREIGN KEY (IdPlci) REFERENCES dict.tbl\_plec(IdPlci),

CONSTRAINT FK\_klient\_kraj FOREIGN KEY (IdKraju) REFERENCES dict.tbl\_kraj(IdKraju)

);

--tabela samochód

CREATE TABLE tbl\_samochod

(

IdSamochodu int NOT NULL PRIMARY KEY,

Rejestracja varchar(15),

IdModelu int NOT NULL,

IdMarki int NOT NULL,

IdRodzajuNadwozia tinyint NOT NULL,

IdSilnika tinyint NOT NULL,

IdRodzajuSamochodu tinyint NOT NULL,

Kolor varchar(30) NOT NULL,

Przebieg int NOT NULL,

Rocznik int NOT NULL,

VIN varchar(17) NOT NULL,

CONSTRAINT FK\_samochod\_model FOREIGN KEY (IdModelu) REFERENCES dict.tbl\_model,

CONSTRAINT FK\_samochod\_marka FOREIGN KEY (IdMarki) REFERENCES dict.tbl\_marka,

CONSTRAINT FK\_samochod\_rodzaj\_nadwozia FOREIGN KEY (IdRodzajuNadwozia) REFERENCES dict.tbl\_rodzaj\_nadwozia,

CONSTRAINT FK\_samochod\_silnik FOREIGN KEY (IdSilnika) REFERENCES dict.tbl\_silnik,

CONSTRAINT FK\_samochod\_rodzaj\_samochodu FOREIGN KEY (IdRodzajuSamochodu) REFERENCES dict.tbl\_rodzaj\_samochodu

);

--tabela ubezpieczenie

CREATE TABLE tbl\_ubezpieczenie

(

IdUbezpieczenia int NOT NULL,

IdSamochodu int NOT NULL,

NumerPolisy varchar(16) NOT NULL,

DataRozpoczeciaUbezpieczenia date NOT NULL,

DataZakonczeniaUbezpieczenia date NOT NULL,

FirmaUbezpieczeniowa varchar(50) NOT NULL,

Cena money NOT NULL,

PRIMARY KEY(IdUbezpieczenia, IdSamochodu),

CONSTRAINT FK\_ubezpieczenie\_samochod FOREIGN KEY (IdSamochodu) REFERENCES tbl\_samochod(IdSamochodu)

);

CREATE TABLE tbl\_wypozyczenie

(

IdWypozyczenia int NOT NULL PRIMARY KEY,

DataWypozyczenia date NOT NULL,

DataZwrotu date NOT NULL,

StanTechniczny varchar(100)

);

CREATE TABLE tbl\_rezerwacja

(

IdRezerwacji int NOT NULL PRIMARY KEY,

IdWypozyczenia int NOT NULL,

IdSamochodu int NOT NULL,

CenaZaDobe money NOT NULL,

Kaucja money NOT NULL,

DataRezerwacji date NOT NULL,

DataKoncaRezerwacji date NOT NULL,

DataUtworzeniaRezerwacji date NOT NULL,

LacznaKwotaDoZaplaty money NOT NULL,

CONSTRAINT FK\_rezerwacja\_wypozyczenie FOREIGN KEY(IdWypozyczenia) REFERENCES tbl\_wypozyczenie(IdWypozyczenia),

CONSTRAINT FK\_rezerwacja\_samochod FOREIGN KEY(IdSamochodu) REFERENCES tbl\_samochod(IdSamochodu)

);

CREATE TABLE tbl\_faktura

(

IdFaktury int NOT NULL PRIMARY KEY,

Tytyl varchar(50) NOT NULL,

Wydawca varchar(30) NOT NULL,

DataWystawienia date NOT NULL,

NIP varchar(11),

NazwaFirmy varchar(50),

NrKontaBankowego varchar(28),

IdRezerwacji int NOT NULL,

IdWypozyczenia int NOT NULL,

IdFormyFaktury tinyint NOT NULL,

CONSTRAINT FK\_faktura\_rezerwacja FOREIGN KEY(IdRezerwacji) REFERENCES tbl\_rezerwacja(IdRezerwacji),

CONSTRAINT FK\_faktura\_wypozyczenie FOREIGN KEY(IdWypozyczenia) REFERENCES tbl\_wypozyczenie(IdWypozyczenia),

CONSTRAINT FK\_faktura\_forma\_faktury FOREIGN KEY(IdFormyFaktury) REFERENCES dict.tbl\_forma\_faktury(IdFormyFaktury)

);

CREATE TABLE tbl\_platnosc

(

IdPlatnosci int NOT NULL PRIMARY KEY,

Zaplata money NOT NULL,

IdFaktury int NOT NULL,

IdFormaPlatnosci tinyint NOT NULL

CONSTRAINT FK\_platnosc\_faktura FOREIGN KEY(IdFaktury) REFERENCES tbl\_faktura(IdFaktury),

CONSTRAINT FK\_platnosc\_forma\_platnosci FOREIGN KEY(IdFormaPlatnosci) REFERENCES dict.tbl\_forma\_platnosci

);

CREATE TABLE tbl\_rezerwacja\_klient

(

IdKlienta int NOT NULL,

IdRezerwacji int NOT NULL,

CONSTRAINT FK\_rezerwacja\_klient\_klient FOREIGN KEY(IdKlienta) REFERENCES tbl\_klient(IdKlienta),

CONSTRAINT FK\_rezerwacja\_klient\_rezerewacja FOREIGN KEY(IdRezerwacji) REFERENCES tbl\_rezerwacja(IdRezerwacji)

);

CREATE TABLE tbl\_spis\_napraw

(

IdNaprawy int NOT NULL PRIMARY KEY,

OpisNaprawy varchar(50) NOT NULL,

DataNaprawy date NOT NULL,

Cena money NOT NULL,

IdSamochodu int NOT NULL,

CONSTRAINT FK\_spis\_napraw\_samochod FOREIGN KEY(IdSamochodu) REFERENCES tbl\_samochod(IdSamochodu)

);

CREATE TABLE tbl\_przeglad\_samochodu

(

IdPrzegladu int NOT NULL PRIMARY KEY,

DataZrobieniaPrzegladu date NOT NULL,

DataWaznosciPrzegladu date NOT NULL,

Uwagi varchar(60),

MiejsceWykonaniaPrzegladu varchar(30) NOT NULL,

Cena money NOT NULL,

IdSamochodu int NOT NULL,

CONSTRAINT FK\_przeglad\_samochod FOREIGN KEY(IdSamochodu) REFERENCES tbl\_samochod(IdSamochodu)

);

# Wykaz zaimplementowanych indeksów

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst, wewnątrz

Opis wygenerowany automatycznieObraz zawierający tekst

Opis wygenerowany automatycznieObraz zawierający tekst

Opis wygenerowany automatycznie



Obraz zawierający tekst

Opis wygenerowany automatycznie

















# Skrypt tworzenia indeksów

CREATE INDEX idx\_klient\_nazwisko

ON dbo.tbl\_klient(Nazwisko);

CREATE INDEX idx\_rezerwacja\_cenazadobe

ON dbo.tbl\_rezerwacja(CenaZaDobe);

CREATE INDEX idx\_klient\_credentials

ON dbo.tbl\_klient(Imie,Nazwisko);

CREATE INDEX idx\_marka\_nazwamarki

ON dict.tbl\_marka(Marka);

CREATE INDEX idx\_model\_nazwamodelu

ON dict.tbl\_model(Model);

CREATE INDEX idx\_wypozyczenie\_datawypozyczenia

ON dbo.tbl\_wypozyczenie(DataWypozyczenia);

CREATE INDEX idx\_ubezpieczenie\_nrpolisy

ON dbo.tbl\_ubezpieczenie(NumerPolisy);

CREATE INDEX idx\_ubezpieczenie\_cena

ON dbo.tbl\_ubezpieczenie(Cena);

CREATE INDEX idx\_klient\_miejscowosc

ON dbo.tbl\_klient(Miejscowosc)

CREATE INDEX idx\_klient\_email

ON dbo.tbl\_klient(Email)

CREATE INDEX idx\_samochod\_idmarki

ON dbo.tbl\_samochod(IdMarki)

CREATE INDEX idx\_faktura\_idrezerwacji

ON dbo.tbl\_faktura(IdRezerwacji)

CREATE INDEX idx\_faktura\_data\_wystawienia\_faktury

ON dbo.tbl\_faktura(DataWystawienia)

CREATE INDEX idx\_faktura\_tytul

ON dbo.tbl\_faktura(Tytyl)

CREATE INDEX idx\_rezerwacja\_idwypozyczenia

ON dbo.tbl\_rezerwacja(IdWypozyczenia)

CREATE INDEX idx\_klient\_kodpocztowy

ON dbo.tbl\_klient(KodPocztowy)

CREATE INDEX idx\_platnosc\_idfaktury

ON dbo.tbl\_platnosc(IdFaktury)

CREATE INDEX idx\_przeglad\_idsamochodu

ON dbo.tbl\_przeglad\_samochodu(IdSamochodu)

CREATE INDEX idx\_rezerwacja\_data\_rezerwacji

ON dbo.tbl\_rezerwacja(DataRezerwacji)

CREATE INDEX idx\_rezerwacja\_laczna\_kwota\_do\_zaplaty

ON dbo.tbl\_rezerwacja(LacznaKwotaDoZaplaty)

CREATE INDEX idx\_samochod\_rocznik

ON dbo.tbl\_samochod(Rocznik)

CREATE INDEX idx\_samochod\_przebieg

ON dbo.tbl\_samochod(Przebieg)

# Skrypt dodający przykładowe dane do tabel

INSERT INTO dict.tbl\_forma\_faktury(IdFormyFaktury,FormaFaktury)

VALUES (1, 'papierowa'), (2, 'elektroniczna');

INSERT INTO dict.tbl\_forma\_platnosci(IdFormyPlatnosci, FormaPlatnosci)

VALUES (1, 'gotówka'), (2, 'przelew'), (3, 'BLIK'), (4, 'karta');

INSERT INTO dict.tbl\_kraj(IdKraju, Kraj)

VALUES (1, 'Austria'),

(2, 'Belgia'),

(3, 'Bułgaria'),

(4, 'Chorwacja'),

(5, 'Cypr'),

(6, 'Czechy'),

(7, 'Dania'),

(8, 'Estonia'),

(9, 'Finlandia'),

(10, 'Francja'),

(11, 'Grecja'),

(12, 'Hiszpania'),

(13, 'Holandia'),

(14, 'Irlandia'),

(15, 'Litwa'),

(16, 'Luksemburg'),

(17, 'Łotwa'),

(18, 'Malta'),

(19, 'Niemcy'),

(20, 'Polska'),

(21, 'Portugalia'),

(22, 'Rumunia'),

(23, 'Słowacja'),

(24, 'Słowenia'),

(25, 'Szwecja'),

(26, 'Węgry'),

(27, 'Włochy');

INSERT INTO dict.tbl\_marka(IdMarki, Marka)

VALUES(1, 'Audi'),

(2, 'BMW'),

(3, 'Chevrolet'),

(4, 'Fiat'),

(5, 'Peugeot'),

(6, 'Porsche'),

(7, 'Seat'),

(8, 'Dacia'),

(9, 'Volkswagen'),

(10, 'Skoda'),

(11, 'Saab'),

(12, 'Mercedes-Benz');

INSERT INTO dict.tbl\_model(IdModelu, Model)

VALUES(1, 'A6'), (2, 'A4'), (3, 'Q3'), (4,'X3'), (5, 'Z3'), (6, 'Aveo'), (7, 'Cruze'), (8, 'Tipo'), (9, 'Panda City Life'), (10, '500'),

(11, '2008'), (12, '208'), (13, '911'), (14, 'Leon'), (15, 'Ateca'), (16, 'Ibiza'), (17, 'Logan'), (18, 'Sandero'), (19, 'Golf'), (20, 'Passat'), (21, 'Polo'),

(22, 'Scirocco'), (23, 'Fabia'), (24, 'Yeti'), (25, 'Octavia'), (26, '9-3'), (27, 'S205'), (28, 'W247');

INSERT INTO dict.tbl\_wojewodztwo

VALUES

(0,'zagranica'),

(1,'dolnośląskie'),

(2,'kuajwsko-pomorksie'),

(3,'lubelskie'),

(4,'lubuskie'),

(5,'łódzkie'),

(6,'małopolskie'),

(7,'mazowieckie'),

(8,'opolskie'),

(9,'podkarpackie'),

(10,'podlaskie'),

(11,'pomorskie'),

(12,'śląskie'),

(13,'świętokrzyskie'),

(14,'warmińsko-mazurskie'),

(15,'wielkopolskie'),

(16,'zachodniopomorskie');

INSERT INTO dict.tbl\_plec

VALUES

(1, 'kobieta','K'),

(2,'mężczyzna','M'),

(3,'inna','I');

INSERT INTO dict.tbl\_rodzaj\_nadwozia

VALUES

(1, 'sedan'),

(2, 'kombi'),

(3, 'coupe'),

(4, 'hatchback'),

(5, ‘suv’);

INSERT INTO dict.tbl\_rodzaj\_samochodu

VALUES

(1,'premium',200),

(2,'standard',100),

(3,'business',150);

INSERT INTO dict.tbl\_silnik

VALUES

(1,'benzynowy'),

(2,'elektryczny'),

(3,'diesel'),

(4,'hybrydowy');

INSERT INTO dbo.tbl\_klient

VALUES(1, 'Mariejeanne', 'Dincey', '2098 Monterey Trail', 'Warszawa', '22-444', 7, 20, '+86-594-222-348', 'mdincey@sda.pl', '1982-10-24', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(2, 'Issy', 'Rubertelli', '695 Melvin Junction', 'Ateny', '05-457', 0, 11, '+57-805-544-817', 'issy2231@shm.gre','1994-06-02', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(3, 'Bethanne', 'Linny', '1416 Bluejay Place', 'Bukareszt', '96-710', 0, 22, '+359-594-845-92', 'betlin70@noaa.gov', '1970-02-21', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(4, 'Tildie', 'Lovelock', '13753 Nancy Court', 'Dublin', '00-241', 0, 14, '+86-477-271-657', 'mdincey@sidemeter.org', '1962-12-24', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(5, 'Cletus', 'Wellard', '90 Green Ridge Street', 'Dublin', '00-990', 0, 14, '+86-288-564-177', 'mdincey@marriot.com','1994-01-15', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(6, 'Thacher', 'Yurkov', '7 Sheridan Court', 'Łódź', '93-112', 5, 20, '+1-313-259-837', 'mdincey@polmat.pl','1998-07-02', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(7, 'Megen', 'Leasor', '9427 Roxbury Road', 'Sofia', '11-192', 0, 3, '+80-923-375-373','megenleasor@ger.com', '2000-09-24', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(8, 'Belinda', 'Agney', '84607 Moose Hill', 'Berlin', '22-122', 0, 19, '+59-416-467-422','belindaagney@aabb.com', '1978-08-29', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(9, 'Crin', 'Pedrielli', '6487 Mccormick Road', 'Turyn', '44-765' , 0, 27 , '+82-395-592-651', 'cpedriellia@cnet.com', '1990-04-10', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(10, 'Oralie', 'MacKee', '780 Sutteridge Trail', 'Monachium', '02-129', 0, 19, '+86-920-130-748', 'omackeeb@topsy.com', '1955-02-02', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(11, 'Tricia', 'Lamberteschi', '02866 Victoria Center', 'Poznań', '95-442', 15, 20, '+1-254-277-256', 'tlamberteschic@intel.com', '1980-04-20', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(12, 'Mignonne', 'Coen', '44 Heffernan Court', 'Warszawa', '90-111', 7, 20, '+86-664-171-826', 'mcoend@instagram.com', '1972-04-12', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(13, 'Dionysus', 'Frango', '788 John Wall Junction', 'Kraków', '95-231', 6, 20, '+38-456-314-838', 'dfrangoe@jalbum.net', '1974-11-11', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(14, 'Danella', 'Mincher', '97 Annamark Court', 'Kielce', '90-023', 13, 20, '+23-704-904-390', 'dmincherf@nationalgeographic.com', '1994-05-10', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(15, 'Cass', 'Theml', '12294 Cascade Terrace', 'Wilno', '35-900', 0, 15, '+7-840-762-968', 'cthemlg@feedburner.com', '1992-07-29', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(16, 'Alanna', 'Massard', '1 Kinsman Plaza', 'Gdańsk', '85-201', 11, 20, '+35-818-618-109', 'amassardh@oakley.com', '2000-03-31', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(17, 'Jewell', 'Rosiello', '06 Sunfield Trail', 'Radom', '99-023', 7, 20, '+7-386-809-41', 'jrosiellop@xinhuanet.com', '1968-12-04', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(18, 'Wilt', 'Trevarthen', '0037 Hauk Lane', 'Częstochowa', '54-525', 12, 20, '+46-797-401-73', 'wtrevarthenq@opensource.org', '1971-11-24', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(19, 'Anthony', 'Alfwy', '39138 Eggendart Avenue', 'Sztokholm', '02-920', 0, 25, '+86-293-686-85', 'aalfwyr@illinois.edu', '1989-09-02', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(20, 'Ximenez', 'Roslen', '70161 Tomscot Terrace', 'Helsinki', '67-221', 0, 9, '+86-151-526-205', 'xroslens@amazon.com', '1991-08-17', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(21, 'Jone', 'Ludy', '12082 Melvin Hill', 'Paryż', '88-123', 0, 10, '+53-839-985-44', null, '1949-01-08', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(22, 'Lesly', 'Ruggen', '50417 Marquette Terrace', 'Praga', '77-192', 0, 6, '+62-114-718-54', null, '1952-09-01', 2,null)

INSERT INTO dbo.tbl\_klient

VALUES(23, 'Sue', 'Clamo', '37 Shelley Parkway', 'Amsterdam', '12-092', 0, 13, '+86-504-238-64', 'sclamo5@admin.ch', '1968-10-02', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(24, 'Viviene', 'Jirek', '64 Oneill Street', 'Bratysława', '09-820', 0, 23, '+86-637-293-69', 'vjirek6@smugmug.com', '1970-08-21', 1,null)

INSERT INTO dbo.tbl\_klient

VALUES(25, 'Madeleine', 'Blaxland', '641 Lerdahl Center', 'Bruksela', '24-887', 0, 2, '+59-858-640-05', 'mblaxland9@china.com.cn', '1988-03-20', 1,null)

INSERT INTO tbl\_samochod

VALUES

(1,'SL61801',1,1,1,1,1,'czarny',170000,2010,'JAE36BLC0PJC92436'),

(2,'CLI2807',2,1,1,3,2,'srebrny',200000,2009,'1GC316C651F701385'),

(3,'ZK60338',3,1,5,1,1,'brązowy',30000,2019,'KL1G2ETP1NSTZ5758'),

(4,'SPS4407',4,2,5,1,2,'zielony',150000,2010,'4S3WR1WB2RP8K7761'),

(5,'ZGR2955',5,2,3,1,2,'niebieski',120000,2011,'1GB2NEXJ5RG6F4924'),

(6,'LKS4739',6,3,4,1,2,'czerwony',250000,2007,'7A1EERD60XEET3381'),

(7,'RBR2426',7,3,1,3,3,'biały',15000,2020,'WBXMMF0K2VUHV6056'),

(8,'NEB7732',8,4,2,3,2,'srebrny',211000,2009,'1G41T7C535LRW8686'),

(9,'NLI7617',9,4,4,3,2,'różowy',350000,2006,'SCC53U2F940HS5741'),

(10,'PP78938',10,4,4,1,2,'biały',100000,2015,'8AW1SP8T11DJ98287'),

(11,'WH49846',11,5,5,2,3,'fioletowy',200000,2011,'4USJFXL764YEN0616'),

(12,'FZA9211',12,5,4,1,2,'czarny',150000,2015,'4S60LFRC1VESD8536'),

(13,'EZD4988',13,6,3,1,1,'szary',10000,2019,'MMBM0WX875ZPG9233'),

(14,'ZST1820',14,7,4,3,2,'bordowy',250000,2008,'9UK0CJH59JF0N9671'),

(15,'SJ06192',15,7,4,3,2,'zielony',150000,2012,'AHTYTTJZ88Y7S1353'),

(16,'LKR1952',16,7,4,2,2,'biały',20000,2020,'WP0X2DCD9L4WU2485'),

(17,'GKA9207',17,8,2,3,2,'niebieski',150000,2013,'988CWERM96GBH0269'),

(18,'WGS7088',18,8,5,1,2,'pomarańczowy',150000,2012,'PNVSSUAF4XNNU8513'),

(19,'KNS0026',19,9,4,3,2,'czarny',350000,2006,'JC19HZXU9ZVNF5972'),

(20,'FMI3212',20,9,1,3,1,'czarny',100000,2019,'935GH2TX86VJU0737'),

(21,'BBI7875',21,9,4,3,2,'granatowy',350000,2002,'SHSNAWH58A3JS0331'),

(22,'EWI0098',22,9,4,3,2,'biały',200000,2011,'KNKT04H22HPLM3091'),

(23,'PGN4395',23,10,4,1,2,'srebrny',90000,2015,'1NMT7KH68KYUP4978'),

(24,'PCT9078',24,10,2,1,3,'biały',9500,2021,'WJR6VEWH2BYBK5375'),

(25,'RST9602',25,10,1,1,1,'czerwony',10000,2021,'PNVWWLY18S4GT5794'),

(26,'DLW3954',26,11,1,3,2,'granatowy',300000,2006,'3NBLR2S19D1A17793'),

(27,'FWS7901',27,12,1,1,1,'biały',15000,2020,'2HGVHY1302CA49485'),

(28,'DBA9188',28,12,1,1,1,'srebrny',200000,2009,'1HG4UYE24S9E56519')

INSERT INTO tbl\_wypozyczenie

VALUES

(1,'2022-06-06','2022-06-08','dobry'),

(2,'2022-05-12','2022-05-14','zarysowane drzwi pasażera'),

(3,'2022-03-11','2022-03-12','dobry'),

(4,'2022-02-22','2022-03-01','dobry'),

(5,'2022-02-14','2022-02-17','uszkodzone prawe lusterko zewnętrzne'),

(6,'2022-02-01','2022-02-03','dobry'),

(7,'2021-12-29','2022-12-31','urwana klamka od drzwi kierowcy'),

(8,'2021-12-20','2022-12-24','dobry'),

(9,'2021-12-15','2022-12-20','dobry'),

(10,'2021-12-15','2022-12-20','dobry'),

(11,'2021-12-15','2021-12-16','zabrudzona tapiecerka na siedzeniu kierowcy'),

(12,'2021-11-30','2021-12-12','uszkodzona tylna lewa lampa, porysowany tylny zderzak'),

(13,'2021-12-12','2021-12-15','dobry'),

(14,'2022-04-04','2022-04-10','dobry'),

(15,'2022-03-03','2022-03-12','dobry'),

(16,'2022-06-05','2022-06-10','pęknięta tylna szyba'),

(17,'2021-11-12','2021-12-01','dobry'),

(18,'2022-05-09','2022-06-09','dobry'),

(19,'2021-10-29','2021-11-05','dobry'),

(20,'2022-06-06','2022-06-10','zarysowany przedni lewy błotnik'),

(21,'2022-06-15','2022-06-20','dobry')

INSERT INTO tbl\_ubezpieczenie

VALUES

(NEXT VALUE FOR dbo.seq\_new\_id,1,'4436503045195625','2022-01-01','2023-01-01','Warta',800),

(NEXT VALUE FOR dbo.seq\_new\_id,2,'0662675557993847','2022-05-12','2023-05-12','ErgoHestia',800),

(NEXT VALUE FOR dbo.seq\_new\_id,3,'2737877329385536','2022-04-04','2023-04-04','Warta',899),

(NEXT VALUE FOR dbo.seq\_new\_id,4,'2099744314252624','2021-12-12','2022-12-12','Generalii',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,5,'4623306699555827','2021-11-04','2022-11-04','Warta',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,6,'9166510753280534','2022-01-01','2023-01-01','Warta',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,7,'1575641647196365','2021-06-30','2022-06-30','Warta',1500),

(NEXT VALUE FOR dbo.seq\_new\_id,8,'4320528151956208','2022-03-20','2023-03-20','Warta',1500),

(NEXT VALUE FOR dbo.seq\_new\_id,9,'8834250622586231','2021-07-07','2022-07-07','ErgoHestia',2000),

(NEXT VALUE FOR dbo.seq\_new\_id,10,'6521803312985063','2022-01-01','2023-01-01','Warta',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,11,'8981597713686992','2021-06-30','2022-06-30','Warta',1200),

(NEXT VALUE FOR dbo.seq\_new\_id,12,'5143277307257895','2021-07-01','2022-07-01','AXA',900),

(NEXT VALUE FOR dbo.seq\_new\_id,13,'8763741978747673','2021-06-26','2022-06-26','AXA',2600),

(NEXT VALUE FOR dbo.seq\_new\_id,14,'8160944016585544','2022-05-12','2023-05-12','Warta',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,15,'3080599376753492','2022-01-01','2023-01-01','Warta',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,16,'6636076142997430','2021-10-12','2022-10-12','Generali',1200),

(NEXT VALUE FOR dbo.seq\_new\_id,17,'5100103960505421','2021-12-16','2022-12-16','Warta',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,18,'0670332762736141','2022-06-06','2023-06-06','Warta',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,19,'2524554269224534','2021-08-01','2022-08-01','Warta',2000),

(NEXT VALUE FOR dbo.seq\_new\_id,20,'1155583100572995','2022-05-26','2023-05-26','Warta',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,21,'0873171488970813','2021-09-01','2021-09-01','Warta',870),

(NEXT VALUE FOR dbo.seq\_new\_id,22,'1498992752251132','2022-12-01','2023-12-01','Warta',760),

(NEXT VALUE FOR dbo.seq\_new\_id,23,'9160975865574716','2021-11-09','2022-11-09','Warta',1200),

(NEXT VALUE FOR dbo.seq\_new\_id,24,'1040968581002422','2022-04-04','2023-04-04','Warta',1020),

(NEXT VALUE FOR dbo.seq\_new\_id,25,'5934692348085146','2021-07-01','2022-07-01','Warta',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,26,'2972567478502828','2022-05-05','2023-05-05','ErgoHestia',500),

(NEXT VALUE FOR dbo.seq\_new\_id,27,'7524557938780253','2021-06-25','2022-06-25','Warta',1000),

(NEXT VALUE FOR dbo.seq\_new\_id,28,'1257832617645923','2022-02-02','2023-02-02','ErgoHestia',1200)

ALTER SEQUENCE dbo.seq\_new\_id RESTART;

INSERT INTO tbl\_spis\_napraw

VALUES

(NEXT VALUE FOR dbo.seq\_new\_id,'Wypolerowano lakier na drzwiach','2022-05-20',350,12),

(NEXT VALUE FOR dbo.seq\_new\_id,'Wymieniono prawe lusterko na nowe','2022-02-25',450,2),

(NEXT VALUE FOR dbo.seq\_new\_id,'Wymiana klamki od drzwi kierowcy','2022-01-12',200,3),

(NEXT VALUE FOR dbo.seq\_new\_id,'Wyprano środek samochodu','2021-12-20',150,7),

(NEXT VALUE FOR dbo.seq\_new\_id,'Wymieniono lampe i zderzak','2021-12-20',900,14),

(NEXT VALUE FOR dbo.seq\_new\_id,'Wymieniono tylną szybę','2022-06-12',600,11),

(NEXT VALUE FOR dbo.seq\_new\_id,'Wymieniono lewy przedni błotnik','2022-06-14',700,11)

ALTER SEQUENCE dbo.seq\_new\_id RESTART;

INSERT INTO tbl\_przeglad\_samochodu

VALUES

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=1),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=1),'','S.K.P w Pleszewie',100,1),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=2),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=2),'wyciek z tylnego dyferyncjału','S.K.P w Pleszewie',100,2),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=3),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=3),'','S.K.P w Pleszewie',100,3),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=4),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=4),'','S.K.P w Pleszewie',100,4),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=5),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=5),'','S.K.P w Pleszewie',100,5),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=6),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=6),'luzy w zawieszeniu','S.K.P w Pleszewie',100,6),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=7),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=7),'','S.K.P w Pleszewie',100,7),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=8),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=8),'','S.K.P w Pleszewie',100,8),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=9),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=9),'','S.K.P w Pleszewie',100,9),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=10),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=10),'hamulce nadają się do wymiany','S.K.P w Pleszewie',100,10),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=11),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=11),'','S.K.P w Pleszewie',100,11),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=12),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=12),'','S.K.P w Pleszewie',100,12),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=13),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=13),'wyciek ze skrzyni biegów','S.K.P w Pleszewie',100,13),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=14),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=14),'','S.K.P w Pleszewie',100,14),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=15),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=15),'','S.K.P w Pleszewie',100,15),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=16),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=16),'','S.K.P w Pleszewie',100,16),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=17),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=17),'opony nadają się do wymiany','S.K.P w Pleszewie',100,17),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=18),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=18),'','S.K.P w Pleszewie',100,18),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=19),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=19),'','S.K.P w Pleszewie',100,19),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=20),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=20),'','S.K.P w Pleszewie',100,20),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=21),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=21),'luzy w poduszcze silnika','S.K.P w Pleszewie',100,21),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=22),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=22),'','S.K.P w Pleszewie',100,22),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=23),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=23),'','S.K.P w Pleszewie',100,23),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=24),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=24),'','S.K.P w Pleszewie',100,24),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=25),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=25),'','S.K.P w Pleszewie',100,25),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=26),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=26),'należy wymienić przedni lewy wahacz','S.K.P w Pleszewie',100,26),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=27),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=27),'','S.K.P w Pleszewie',100,27),

(NEXT VALUE FOR dbo.seq\_new\_id,(SELECT DataRozpoczeciaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=28),(SELECT DataZakonczeniaUbezpieczenia FROM tbl\_ubezpieczenie WHERE IdSamochodu=28),'','S.K.P w Pleszewie',100,28)

INSERT INTO tbl\_rezerwacja

VALUES

(1,1,1,dbo.uf\_getCar(1),1000,dbo.uf\_getDataWypozyczenia(1),dbo.uf\_getDataZwrotu(1),'2022-06-05',(dbo.uf\_czyZwrocicKaucje(1)+CONVERT(money,dbo.uf\_getCar(1)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(1),dbo.uf\_getDataZwrotu(1))))),

(2,2,2,dbo.uf\_getCar(2),1000,dbo.uf\_getDataWypozyczenia(2),dbo.uf\_getDataZwrotu(2),'2022-05-11',(dbo.uf\_czyZwrocicKaucje(2)+CONVERT(money,dbo.uf\_getCar(2)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(2),dbo.uf\_getDataZwrotu(2))))),

(3,3,4,dbo.uf\_getCar(4),1000,dbo.uf\_getDataWypozyczenia(3),dbo.uf\_getDataZwrotu(3),'2022-03-08',(dbo.uf\_czyZwrocicKaucje(3)+CONVERT(money,dbo.uf\_getCar(4)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(3),dbo.uf\_getDataZwrotu(3))))),

(4,4,5,dbo.uf\_getCar(5),1000,dbo.uf\_getDataWypozyczenia(4),dbo.uf\_getDataZwrotu(4),'2022-02-12',(dbo.uf\_czyZwrocicKaucje(4)+CONVERT(money,dbo.uf\_getCar(5)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(4),dbo.uf\_getDataZwrotu(4))))),

(5,5,6,dbo.uf\_getCar(6),1000,dbo.uf\_getDataWypozyczenia(5),dbo.uf\_getDataZwrotu(5),'2022-01-30',(dbo.uf\_czyZwrocicKaucje(5)+CONVERT(money,dbo.uf\_getCar(6)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(5),dbo.uf\_getDataZwrotu(5))))),

(6,6,7,dbo.uf\_getCar(7),1000,dbo.uf\_getDataWypozyczenia(6),dbo.uf\_getDataZwrotu(6),'2021-12-25',(dbo.uf\_czyZwrocicKaucje(6)+CONVERT(money,dbo.uf\_getCar(7)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(6),dbo.uf\_getDataZwrotu(6))))),

(7,7,8,dbo.uf\_getCar(8),1000,dbo.uf\_getDataWypozyczenia(7),dbo.uf\_getDataZwrotu(7),'2021-12-18',(dbo.uf\_czyZwrocicKaucje(7)+CONVERT(money,dbo.uf\_getCar(8)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(7),dbo.uf\_getDataZwrotu(7))))),

(8,8,10,dbo.uf\_getCar(10),1000,dbo.uf\_getDataWypozyczenia(8),dbo.uf\_getDataZwrotu(8),'2021-12-13',(dbo.uf\_czyZwrocicKaucje(8)+CONVERT(money,dbo.uf\_getCar(10)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(8),dbo.uf\_getDataZwrotu(8))))),

(9,9,11,dbo.uf\_getCar(11),1000,dbo.uf\_getDataWypozyczenia(9),dbo.uf\_getDataZwrotu(9),'2021-12-13',(dbo.uf\_czyZwrocicKaucje(9)+CONVERT(money,dbo.uf\_getCar(11)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(9),dbo.uf\_getDataZwrotu(9))))),

(10,10,9,dbo.uf\_getCar(9),1000,dbo.uf\_getDataWypozyczenia(10),dbo.uf\_getDataZwrotu(10),'2021-12-14',(dbo.uf\_czyZwrocicKaucje(10)+CONVERT(money,dbo.uf\_getCar(9)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(10),dbo.uf\_getDataZwrotu(10))))),

(11,11,12,dbo.uf\_getCar(12),1000,dbo.uf\_getDataWypozyczenia(11),dbo.uf\_getDataZwrotu(11),'2021-12-12',(dbo.uf\_czyZwrocicKaucje(11)+CONVERT(money,dbo.uf\_getCar(12)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(11),dbo.uf\_getDataZwrotu(11))))),

(12,12,13,dbo.uf\_getCar(13),1000,dbo.uf\_getDataWypozyczenia(12),dbo.uf\_getDataZwrotu(12),'2021-11-26',(dbo.uf\_czyZwrocicKaucje(12)+CONVERT(money,dbo.uf\_getCar(13)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(12),dbo.uf\_getDataZwrotu(12))))),

(13,13,14,dbo.uf\_getCar(14),1000,dbo.uf\_getDataWypozyczenia(13),dbo.uf\_getDataZwrotu(13),'2021-12-08',(dbo.uf\_czyZwrocicKaucje(13)+CONVERT(money,dbo.uf\_getCar(14)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(13),dbo.uf\_getDataZwrotu(13))))),

(14,14,15,dbo.uf\_getCar(15),1000,dbo.uf\_getDataWypozyczenia(14),dbo.uf\_getDataZwrotu(14),'2022-04-01',(dbo.uf\_czyZwrocicKaucje(14)+CONVERT(money,dbo.uf\_getCar(15)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(14),dbo.uf\_getDataZwrotu(14))))),

(15,15,1,dbo.uf\_getCar(1),1000,dbo.uf\_getDataWypozyczenia(15),dbo.uf\_getDataZwrotu(15),'2022-03-02',(dbo.uf\_czyZwrocicKaucje(15)+CONVERT(money,dbo.uf\_getCar(1)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(15),dbo.uf\_getDataZwrotu(15))))),

(16,16,2,dbo.uf\_getCar(2),1000,dbo.uf\_getDataWypozyczenia(16),dbo.uf\_getDataZwrotu(16),'2022-06-04',(dbo.uf\_czyZwrocicKaucje(16)+CONVERT(money,dbo.uf\_getCar(2)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(16),dbo.uf\_getDataZwrotu(16))))),

(17,17,22,dbo.uf\_getCar(22),1000,dbo.uf\_getDataWypozyczenia(17),dbo.uf\_getDataZwrotu(17),'2021-11-09',(dbo.uf\_czyZwrocicKaucje(17)+CONVERT(money,dbo.uf\_getCar(22)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(17),dbo.uf\_getDataZwrotu(17))))),

(18,18,20,dbo.uf\_getCar(20),1000,dbo.uf\_getDataWypozyczenia(18),dbo.uf\_getDataZwrotu(18),'2022-05-08',(dbo.uf\_czyZwrocicKaucje(18)+CONVERT(money,dbo.uf\_getCar(20)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(18),dbo.uf\_getDataZwrotu(18))))),

(19,19,18,dbo.uf\_getCar(18),1000,dbo.uf\_getDataWypozyczenia(19),dbo.uf\_getDataZwrotu(19),'2021-10-28',(dbo.uf\_czyZwrocicKaucje(19)+CONVERT(money,dbo.uf\_getCar(18)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(19),dbo.uf\_getDataZwrotu(19))))),

(20,20,16,dbo.uf\_getCar(16),1000,dbo.uf\_getDataWypozyczenia(20),dbo.uf\_getDataZwrotu(20),'2022-06-05',(dbo.uf\_czyZwrocicKaucje(20)+CONVERT(money,dbo.uf\_getCar(16)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(20),dbo.uf\_getDataZwrotu(20))))),

(21,21,16,dbo.uf\_getCar(16),1000,dbo.uf\_getDataWypozyczenia(21),dbo.uf\_getDataZwrotu(21),'2022-06-15',(dbo.uf\_czyZwrocicKaucje(21)+CONVERT(money,dbo.uf\_getCar(16)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(21),dbo.uf\_getDataZwrotu(21)))))

INSERT INTO tbl\_rezerwacja\_klient

VALUES

(1,1),

(2,2),

(3,3),

(4,4),

(5,5),

(6,6),

(7,7),

(8,8),

(9,9),

(10,10),

(11,11),

(12,12),

(13,13),

(14,14),

(15,15),

(16,16),

(17,17),

(18,18),

(19,19),

(20,20)

 INSERT INTO tbl\_faktura

VALUES

(1,'Faktura nr 1/1563/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(1),'3784313982','ABC','92109014218305896215261828',1,1,1),

(2,'Faktura nr 2/8606/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(2),'5677862347','Deflex','17102027337012704822330779',2,2,1),

(3, 'Faktura nr 3/2524/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(3),'9328981251','Huragan','74817000080955777312497210',3,3,1),

(4, 'Faktura nr 4/7513/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(4),'1545850629','Wisła','72109013756778831668846245',4,4,2),

(5, 'Faktura nr 5/7937/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(5),'5263242714','Kanał Sportowy','47124049666273325535803775',5,5,2),

(6, 'Faktura nr 6/5011/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(6),'5125600738','Meczyki','28823300044238668504428842',6,6,1),

(7, 'Faktura nr 7/6958/2021','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(7),'9714518179','Trawnik','50948411633777375061893014',7,7,2),

(8, 'Faktura nr 8/9851/2021','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(8),'3560531473','MB Promotion','23897800084249739659965058',8,8,1),

(9, 'Faktura nr 9/4142/2021','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(9),'8235879849','FAME MMA','48904210263824782910738141',9,9,2),

(10, 'Faktura nr 10/8591/2021','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(10),'1589301305','High League','69150023544039376819501367',10,10,1),

(11, 'Faktura nr 11/1269/2021','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(11),'5114949437','Polsport','52124029918099003251137009',11,11,2),

(12, 'Faktura nr 12/8725/2021','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(12),'1250867752','PiS','38137013695080241286266264',12,12,1),

(13, 'Faktura nr 13/7623/2021','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(13),'9519609605','Gazeta Wyborcza','75909400061355586558171290',13,13,2),

(14, 'Faktura nr 14/6343/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(14),'8373590205','Przegląd Sportowy','65132017979161375809408650',14,14,1),

(15, 'Faktura nr 15/3013/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(15),'3947766949','Newonce','94846700017598833699765286',15,15,2),

(16, 'Faktura nr 16/9653/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(16),'5297514232','WP','61958310224355038905751863',16,16,1),

(17, 'Faktura nr 17/8956/2021','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(17),'5125893290','Prawda Futbolu','11109014631576011139194714',17,17,2),

(18, 'Faktura nr 18/5576/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(18),'8224223223','Onet','50804610122548129962271793',18,18,2),

(19, 'Faktura nr 19/7355/2021','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(19),'9715898250','sport.pl','39114021213378591155973913',19,19,2),

(20, 'Faktura nr 20/1819/2022','Wypozyczalnia S.A',dbo.uf\_getDataZwrotu(20),'5194427208','Poboty','04124018776491453670892586',20,20,1)

ALTER SEQUENCE dbo.seq\_new\_id RESTART;

INSERT INTO tbl\_platnosc

VALUES

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(1), 1, 4),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(2), 2, 2),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(3), 3, 2),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(4), 4, 2),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(5), 5, 1),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(6), 6, 3),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(7), 7, 1),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(8), 8, 1),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(9), 9, 2),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(10), 10, 2),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(11), 11, 4),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(12), 12, 4),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(13), 13, 4),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(14), 14, 4),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(15), 15, 3),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(16), 16, 3),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(17), 17, 4),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(18), 18, 4),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(19), 19, 1),

(NEXT VALUE FOR dbo.seq\_new\_id, dbo.uf\_getLacznaKwota(20), 20, 1)

# Wykaz utworzonych sekwencji

Obraz zawierający tekst

Opis wygenerowany automatycznie

# Skrypt tworzący sekwencję

CREATE SEQUENCE seq\_new\_id\_klient

INCREMENT BY 11

START WITH 100

NO CYCLE

GO

CREATE SEQUENCE seq\_new\_id\_samochod

INCREMENT BY 11

START WITH 100

NO CYCLE

GO

CREATE SEQUENCE seq\_new\_id\_rezerwacja

INCREMENT BY 11

START WITH 100

NO CYCLE

GO

CREATE SEQUENCE seq\_new\_id\_wypozyczenie

INCREMENT BY 11

START WITH 100

NO CYCLE

GO

CREATE SEQUENCE seq\_new\_id

INCREMENT BY 10

START WITH 1

NO MINVALUE

NO CYCLE

;

CREATE SEQUENCE seq\_new\_id\_faktury

INCREMENT BY 11

START WITH 100

NO CYCLE

CREATE SEQUENCE seq\_new\_id\_przeglad\_samochod

INCREMENT BY 11

START WITH 100

NO CYCLE

CREATE SEQUENCE seq\_new\_id\_naprawy\_samochod

INCREMENT BY 11

START WITH 100

NO CYCLE

CREATE SEQUENCE seq\_new\_id\_ubezpieczenie

INCREMENT BY 11

START WITH 100

NO CYCLE

CREATE SEQUENCE seq\_new\_id\_platnosci

INCREMENT BY 11

START WITH 100

NO CYCLE

CREATE SEQUENCE seq\_new\_id\_pozycji\_faktury

INCREMENT BY 11

START WITH 100

NO CYCLE

# Wykaz utworzonych widoków

Obraz zawierający tekst

Opis wygenerowany automatycznie

# Skrypt tworzenia widoków

CREATE VIEW vw\_wszyscy\_klienci AS

SELECT \* FROM dbo.tbl\_klient;

CREATE VIEW vw\_klienci\_z\_wojewodztwami\_i\_krajami AS

SELECT IdKlienta, Imie, Nazwisko, Adres, Miejscowosc, w.Wojewodztwo, p.Kraj, pl.Plec FROM dbo.tbl\_klient k

INNER JOIN dict.tbl\_wojewodztwo w ON w.IdWojewodztwa = k.IdWojewodztwa

INNER JOIN dict.tbl\_kraj p ON p.IdKraju = k.IdKraju

INNER JOIN dict.tbl\_plec pl ON pl.IdPlci = k.IdPlci

WITH CHECK OPTION;

CREATE VIEW vw\_klienci\_z\_polski AS

SELECT IdKlienta, Imie, Nazwisko, Adres, Miejscowosc, kraj.Kraj FROM dbo.tbl\_klient klient INNER JOIN dict.tbl\_kraj kraj ON kraj.IdKraju = klient.IdKraju WHERE kraj.Kraj = 'Polska';

CREATE VIEW vw\_marki AS

SELECT \* FROM dict.tbl\_marka;

CREATE VIEW vw\_modele AS

SELECT \* FROM dict.tbl\_model;

CREATE VIEW vw\_rezerwacje\_wszystkie AS

SELECT \* FROM dbo.tbl\_rezerwacja;

CREATE VIEW vw\_rezerwacje\_z\_obecnego\_miesiaca AS

SELECT \* FROM dbo.tbl\_rezerwacja WHERE MONTH(DataRezerwacji) = MONTH(GETDATE());

CREATE VIEW vw\_rezerwacje\_z\_obecnego\_roku AS

SELECT \* FROM dbo.tbl\_rezerwacja WHERE YEAR(DataRezerwacji) = YEAR(GETDATE());

CREATE VIEW vw\_rezerwacje\_z\_zeszłego\_roku AS

SELECT \* FROM dbo.tbl\_rezerwacja WHERE YEAR(DataRezerwacji) < YEAR(GETDATE());

CREATE VIEW vw\_samochody\_wszystkie AS

SELECT \* FROM tbl\_samochod;

CREATE VIEW vw\_samochody\_benzynowe AS

SELECT \* FROM tbl\_samochod samochod INNER JOIN dict.tbl\_silnik silnik ON samochod.IdSilnika = silnik.IdSilnika WHERE silnik.TypSilnika = 'benzynowy';

CREATE VIEW vw\_samochody\_z\_dieslem AS

SELECT IdSamochodu, IdMarki, silnik.TypSilnika FROM tbl\_samochod samochod INNER JOIN dict.tbl\_silnik silnik ON samochod.IdSilnika = silnik.IdSilnika WHERE silnik.TypSilnika = 'diesel';

CREATE VIEW vw\_samochody\_elektryczne AS

SELECT IdSamochodu, IdMarki, silnik.TypSilnika FROM tbl\_samochod samochod INNER JOIN dict.tbl\_silnik silnik ON samochod.IdSilnika = silnik.IdSilnika WHERE silnik.TypSilnika = 'elektryczny';

CREATE VIEW vw\_samochody\_hybrydowe AS

SELECT IdSamochodu, IdMarki, silnik.TypSilnika FROM tbl\_samochod samochod INNER JOIN dict.tbl\_silnik silnik ON samochod.IdSilnika = silnik.IdSilnika WHERE silnik.TypSilnika = 'hybrydowy';

CREATE VIEW vw\_samochody\_kombi AS

SELECT IdSamochodu, IdMarki, nadwozie.RodzajNadwozia FROM tbl\_samochod samochod INNER JOIN dict.tbl\_rodzaj\_nadwozia nadwozie ON samochod.IdRodzajuNadwozia = nadwozie.IdRodzajuNadwozia WHERE nadwozie.RodzajNadwozia = 'kombi';

CREATE VIEW vw\_samochody\_sedan AS

SELECT IdSamochodu, IdMarki, nadwozie.RodzajNadwozia FROM tbl\_samochod samochod INNER JOIN dict.tbl\_rodzaj\_nadwozia nadwozie ON samochod.IdRodzajuNadwozia = nadwozie.IdRodzajuNadwozia WHERE nadwozie.RodzajNadwozia = 'sedan';

CREATE VIEW vw\_samochody\_hatchback AS

SELECT IdSamochodu, IdMarki, nadwozie.RodzajNadwozia FROM tbl\_samochod samochod INNER JOIN dict.tbl\_rodzaj\_nadwozia nadwozie ON samochod.IdRodzajuNadwozia = nadwozie.IdRodzajuNadwozia WHERE nadwozie.RodzajNadwozia = 'hatchback';

GO

CREATE VIEW vw\_liczba\_samochodow\_by\_rodzaj\_silnika AS

SELECT sil.TypSilnika,COUNT(s.IdSilnika) AS 'liczba\_samochodów' FROM tbl\_samochod s INNER JOIN dict.tbl\_silnik sil ON s.IdSilnika = sil.IdSilnika

GROUP BY sil.TypSilnika, s.IdSilnika;

GO

CREATE VIEW vw\_faktura AS

SELECT DISTINCT faktura.IdFaktury,faktura.Tytyl,faktura.DataWystawienia, faktura.Wydawca AS 'Wydawca faktury',

tbl\_klient.Imie+' '+tbl\_klient.Nazwisko AS 'Imie i nazwisko klienta',

tbl\_klient.Adres AS 'Adres',

tbl\_klient.KodPocztowy AS 'Kod pocztowy klienta',

tbl\_klient.Miejscowosc AS 'Miejscowość klienta',

rezerwacja.DataRezerwacji AS 'Data rozpoczęcia rezerwacji',

rezerwacja.DataKoncaRezerwacji AS 'Data końca rezerwacji',

forma.FormaFaktury AS 'Forma faktury',

p.Zaplata AS ‘Do zapłaty’,

fp.FormaPlatnosci AS 'Forma płatności'

FROM tbl\_faktura faktura

INNER JOIN tbl\_rezerwacja rezerwacja ON rezerwacja.IdRezerwacji = faktura.IdFaktury

JOIN tbl\_rezerwacja\_klient rk ON rezerwacja.IdRezerwacji = rk.IdRezerwacji

JOIN tbl\_klient ON rk.IdKlienta = tbl\_klient.IdKlienta

JOIN dict.tbl\_forma\_faktury forma ON faktura.IdFormyFaktury = forma.IdFormyFaktury

JOIN tbl\_platnosc p ON p.IdFaktury = faktura.IdFaktury

JOIN dict.tbl\_forma\_platnosci fp ON fp.IdFormyPlatnosci = p.IdFormaPlatnosci;

GO

CREATE VIEW vw\_rezerwacja\_pelna AS

SELECT r.IdRezerwacji,

k.Imie+' '+k.Nazwisko AS 'Imie i nazwisko klienta',

k.Telefon AS 'Numer telefonu klienta',

s.IdSamochodu AS 'Nr samochodu',

s.Rejestracja AS 'Numer rejestracyjny samochodu',

marka.Marka AS 'Marka samochodu',

model.Model AS 'Model samochodu',

r.DataUtworzeniaRezerwacji AS 'Data utworzenia rezerwacji',

r.DataRezerwacji AS 'Data rozpoczęcia rezerwacji',

r.DataKoncaRezerwacji AS 'Data końca rezerwacji'

FROM tbl\_rezerwacja r

INNER JOIN tbl\_samochod s ON s.IdSamochodu = r.IdSamochodu

JOIN dict.tbl\_marka marka ON marka.IdMarki = s.IdMarki

JOIN dict.tbl\_model model ON model.IdModelu = s.IdModelu

JOIN tbl\_rezerwacja\_klient rk ON rk.IdRezerwacji = r.IdRezerwacji

JOIN tbl\_klient k ON k.IdKlienta = rk.IdKlienta

GO

CREATE VIEW vw\_wypozyczenie\_pelne AS

SELECT

w.IdWypozyczenia AS 'Numer wypożyczenia',

k.Imie+' '+k.Nazwisko AS 'Imie i nazwisko klienta',

k.Telefon AS 'Numer telefonu klienta',

s.IdSamochodu AS 'Nr samochodu',

s.Rejestracja AS 'Numer rejestracyjny samochodu',

marka.Marka AS 'Marka samochodu',

model.Model AS 'Model samochodu',

w.DataWypozyczenia AS 'Data wypożyczenia samochodu',

w.DataZwrotu AS 'Data zwrotu samochodu',

w.StanTechniczny AS 'Stan techniczny pojazdu po zwrocie samochodu'

FROM tbl\_wypozyczenie w

INNER JOIN tbl\_rezerwacja r ON r.IdWypozyczenia = w.IdWypozyczenia

JOIN tbl\_rezerwacja\_klient rk ON r.IdRezerwacji = rk.IdRezerwacji

JOIN tbl\_klient k ON rk.IdKlienta = k.IdKlienta

JOIN tbl\_samochod s ON s.IdSamochodu = r.IdSamochodu

JOIN dict.tbl\_marka marka ON marka.IdMarki = s.IdMarki

JOIN dict.tbl\_model model ON model.IdModelu = s.IdModelu

GO

CREATE VIEW vw\_platnosci AS

SELECT IdPlatnosci, Zaplata,

IdFaktury AS 'Numer faktury',

fp.FormaPlatnosci AS 'Forma płatności'

FROM tbl\_platnosc p

INNER JOIN dict.tbl\_forma\_platnosci fp ON fp.IdFormyPlatnosci = p.IdFormaPlatnosci

GO

CREATE VIEW vw\_suma\_wszystkich\_platnosci AS

SELECT SUM(Zaplata) AS 'Suma wszystkich płatności' FROM tbl\_platnosc;

GO

CREATE VIEW vw\_ubezpieczenie\_pelne AS

SELECT

u.IdUbezpieczenia AS 'Numer ubezpieczenia',

u.NumerPolisy AS 'Numer polisy',

m.Marka AS 'Marka',

model.Model AS 'Model',

u.DataRozpoczeciaUbezpieczenia AS 'Data rozpoczęcia ubezpieczenia',

u.DataZakonczeniaUbezpieczenia AS 'Data zakończenia ubezpieczenia',

u.FirmaUbezpieczeniowa AS 'Firma ubezpieczeniowa',

u.Cena AS 'Kwota do zapłaty za ubezpieczenia'

FROM tbl\_ubezpieczenie u

INNER JOIN tbl\_samochod s ON s.IdSamochodu = u.IdSamochodu

JOIN dict.tbl\_marka m ON s.IdMarki = m.IdMarki

JOIN dict.tbl\_model model ON model.IdModelu = s.IdModelu

GO

CREATE VIEW vw\_spis\_napraw\_pelny AS

SELECT

naprawa.IdNaprawy AS 'Numer naprawy',

m.Marka AS 'Marka',

model.Model AS 'Model',

naprawa.OpisNaprawy AS 'Opis naprawy',

naprawa.DataNaprawy AS 'Data wykonania naprawy',

naprawa.Cena AS 'Kwota do zapłaty'

FROM tbl\_spis\_napraw naprawa

INNER JOIN tbl\_samochod s ON s.IdSamochodu = naprawa.IdSamochodu

JOIN dict.tbl\_marka m ON s.IdMarki = m.IdMarki

JOIN dict.tbl\_model model ON model.IdModelu = s.IdModelu

# Wykaz utworzonych procedur składowanych

Obraz zawierający stół

Opis wygenerowany automatycznie

# Skrypt tworzenia procedur składowanych

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- usp\_GetErrorInfo

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Wyświetla dokładne informacje na temat przechwyconego błędu

---

--- Parametry wejściowe : brak

---

--- Parametry wyjściowe : wypisanie na konsole informacji na temat błędu

---

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE usp\_GetErrorInfo

AS

SELECT

ERROR\_NUMBER() AS ErrorNumber

,ERROR\_SEVERITY() AS ErrorSeverity

,ERROR\_STATE() AS ErrorState

,ERROR\_LINE () AS ErrorLine

,ERROR\_PROCEDURE() AS ErrorProcedure

,ERROR\_MESSAGE() AS ErrorMessage;

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_dodaj\_klienta(@imie varchar(30),@nazwisko varchar(50),@adres varchar(60),@kod\_pocztowy(6),@id\_wojewodztwa int,@id\_kraju int,@telefon varchar(15), @email varchar(40), @data\_urodzenia date, @id\_plec int

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_klient

--- Dodaje nowego klienta do bazy

---

--- Parametry wejściowe :

--- imie klienta, jest to varchar

--- nazwisko, jest to varchar

--- adres, jest to varchar

--- kod pocztowy, jest to varchar

--- id województwa, jest to int

--- id kraju, jest to int

--- telefon, jest to varchar

--- email, jest to varchar

--- data urodzenia, jest to date

--- id płci, jest to int

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE PROCEDURE up\_dodaj\_klienta @imie varchar(30),

@nazwisko varchar(50),

@adres varchar(60),

@miejscowsc varchar(60),

@kod\_pocztowy varchar(6),

@id\_wojewodztwa int,

@id\_kraju int,

@telefon varchar(15),

@email varchar(40),

@data\_urodzenia date,

@id\_plec int

AS

BEGIN TRY

DECLARE @msg varchar(50)

IF @imie IS NULL OR

@nazwisko IS NULL OR

@adres IS NULL OR

@miejscowsc IS NULL OR

@kod\_pocztowy IS NULL OR

@id\_wojewodztwa IS NULL OR

@id\_kraju IS NULL OR

@telefon IS NULL OR

@data\_urodzenia IS NULL OR

@id\_plec IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartości null'

RAISERROR(21,@msg,1)

END

BEGIN

INSERT INTO tbl\_klient(IdKlienta,Imie,Nazwisko,Adres,Miejscowosc,KodPocztowy,IdWojewodztwa,IdKraju,Telefon,Email,DataUrodzenia,IdPlci)

VALUES(NEXT VALUE FOR seq\_new\_id\_klient,@imie,@nazwisko,@adres,@miejscowsc,@kod\_pocztowy,@id\_wojewodztwa,@id\_kraju,@telefon,@email,@data\_urodzenia,@id\_plec);

PRINT('Dodano nowego klienta')

END

END TRY

BEGIN CATCH

EXECUTE usp\_GetErrorInfo

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_dodaj\_auto(@rejestracja varchar(15),@id\_model int,@id\_marka int,@id\_rodzaj\_nadwozia int,@id\_silnik int,@id\_rodzaj\_samochodu int,@kolor varchar(15), @przebieg int, @rocznik int,@vin varchar(17)

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_samochod

--- Dodaje nowy samochód do bazy

---

--- Parametry wejściowe :

--- rejestracja samochodu, jest to varchar

--- id modelu, jest to varchar

--- id marki, jest to varchar

--- id rodzaju nadwozia, jest to varchar

--- id silnika, jest to int

--- id rodzaju nadwozia, jest to int

--- kolor samochodu, jest to varchar

--- przebieg samochodu, jest to varchar

--- rocznik, jest to date

--- vin, jest to varchar

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE PROCEDURE up\_dodaj\_auto

@rejestracja varchar(15),

@id\_model int,

@id\_marka int,

@id\_rodzaj\_nadwozia int,

@id\_silnik int,

@id\_rodzaj\_samochodu int,

@kolor varchar(30),

@przebieg int,

@rocznik int,

@vin varchar(17)

AS

BEGIN TRY

DECLARE @msg varchar(50)

IF @rejestracja IS NULL OR

@id\_model IS NULL OR

@id\_marka IS NULL OR

@id\_rodzaj\_nadwozia IS NULL OR

@id\_silnik IS NULL OR

@id\_rodzaj\_samochodu IS NULL OR

@kolor IS NULL OR

@przebieg IS NULL OR

@rocznik IS NULL OR

@vin IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartości NULL'

RAISERROR(21,@msg,1)

END

ELSE

BEGIN

INSERT INTO tbl\_samochod(IdSamochodu,Rejestracja,IdModelu,IdMarki,IdRodzajuNadwozia,IdSilnika,IdRodzajuSamochodu,Kolor,Przebieg,Rocznik,VIN)

VALUES(NEXT VALUE FOR seq\_new\_id\_samochod,@rejestracja,@id\_model,@id\_marka,@id\_rodzaj\_nadwozia,@id\_silnik,@id\_rodzaj\_samochodu,@kolor,@przebieg,@rocznik,@vin)

PRINT('Pomyślnie dodano nowy samochód do bazy')

END

END TRY

BEGIN CATCH

EXECUTE usp\_GetErrorInfo

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_dodaj\_fakture(@tytul varchar(50),@wydawca varchar(30),@data\_wystawienia date,@nip varchar(11),@nazwa\_firmy varchar(50),@nr\_konta\_bankowego varchar(28),@id\_rezerwacji int,@id\_wypozyczenia int, @id\_formy\_faktury int

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_faktury

--- Dodaje nową fakturę do bazy danych

---

--- Parametry wejściowe :

--- @tytul - tytuł faktury, typ varchar

--- @wydawca - podmiot wystawiający fakture, typ varchar

--- @data\_wystawienia - data wystawienia faktury, typ date

--- @nip - NIP firmy/osoby na którą wystawiana jest faktura , typ varchar

--- @nazwa\_firmy - nazwa firmy, typ varchar

--- @nr\_konta\_bankowego - numer konta bankowego klienta, typ varchar

--- @id\_rezerwacji - numer identyfikacyjny rezerwacji, typ int

--- @id\_wypozyczenia - numer identyfikacyjny wypożyczenia, typ int

--- @id\_formy faktury - numer identyfikacyjny formy faktury, typ int

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomość wypisana na konsoli informująca o braku wypożyczenia o podanym numerze identyfikacyjnym

--- Wiadomość wypisana na konsoli informująca o braku rezerwacji o podanym numerze identyfikacyjnym

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_dodaj\_fakture

@tytul varchar(50),

@wydawca varchar(30),

@data\_wystawienia date,

@nip varchar(11),

@nazwa\_firmy varchar(50),

@nr\_konta\_bankowego varchar(28),

@id\_rezerwacji int,

@id\_wypozyczenia int,

@id\_formy\_faktury int

AS

BEGIN TRY

DECLARE @msg varchar(100)

IF @tytul IS NULL OR

@wydawca IS NULL OR

@data\_wystawienia IS NULL OR

@nip IS NULL OR

@nazwa\_firmy IS NULL OR

@nr\_konta\_bankowego IS NULL OR

@id\_rezerwacji IS NULL OR

@id\_wypozyczenia IS NULL OR

@id\_formy\_faktury IS NULL

BEGIN

SET @msg = 'Kolumny nie przyjmują wartości NULL'

RAISERROR(@msg,16,1)

END

ELSE

BEGIN

IF((SELECT COUNT(IdRezerwacji) FROM tbl\_rezerwacja WHERE IdRezerwacji = @id\_rezerwacji) = 1)

BEGIN

IF((SELECT COUNT(IdWypozyczenia) FROM tbl\_wypozyczenie WHERE IdWypozyczenia = @id\_wypozyczenia) = 1)

BEGIN

INSERT INTO tbl\_faktura(IdFaktury,Tytyl,Wydawca,DataWystawienia,NIP,NazwaFirmy,NrKontaBankowego,IdRezerwacji,IdWypozyczenia,IdFormyFaktury)

VALUES(NEXT VALUE FOR dbo.seq\_new\_id\_faktury,@tytul,@wydawca,@data\_wystawienia,@nip,@nazwa\_firmy,@nr\_konta\_bankowego,@id\_rezerwacji,@id\_wypozyczenia,@id\_formy\_faktury)

END

ELSE

BEGIN

PRINT('Nie znaleziono wypożyczenia o podanym ID')

END

END

ELSE

BEGIN

PRINT('Nie znaleziono rezerwacji o podanym ID')

END

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_dodaj\_naprawe\_samochodu(@opis\_naprawy varchar(50),@data\_naprawy date, @cena money, @id\_samochodu int)

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_spis\_naprawa

--- Dodaje nowy wpis informujący o naprawie danego samochodu

---

--- Parametry wejściowe :

--- @opis\_naprawy - krótki opis naprawy, typ varchar

--- @data\_naprawy - data naprawy samochodu, typ date

--- @cena - cena naprawy, typ money

--- @id\_samochodu - numer identyfikacyjny samochodu, którego tyczy się naprawa , typ varchar

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomość wypisana na konsoli informująca o braku samochodu o podanym numerze identyfikacyjnym

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_dodaj\_naprawe\_samochodu

@opis\_naprawy varchar(50),

@data\_naprawy date,

@cena money,

@id\_samochodu int

AS

BEGIN TRY

DECLARE @msg varchar(100)

IF @opis\_naprawy IS NULL OR

@data\_naprawy IS NULL OR

@cena IS NULL OR

@id\_samochodu IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartość NULL'

RAISERROR(16,@msg,1)

END

ELSE

BEGIN

IF((SELECT COUNT(IdSamochodu) FROM tbl\_samochod WHERE IdSamochodu = @id\_samochodu)=1)

BEGIN

INSERT INTO tbl\_spis\_napraw(IdNaprawy,OpisNaprawy,DataNaprawy,Cena,IdSamochodu)

VALUES(NEXT VALUE FOR seq\_new\_id\_naprawy\_samochod,@opis\_naprawy,@data\_naprawy,@cena,@id\_samochodu)

PRINT('Pomyślnie dodano rekord do tabeli')

END

ELSE

BEGIN

PRINT('W bazie nie ma samochodu o podanym ID!')

END

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_dodaj\_platnosc(@zaplata money,@id\_faktury int, @id\_forma\_platnosci int)

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_platnosc

--- Dodaje nową płatność do bazy

---

--- Parametry wejściowe :

--- @zaplata - kwota do zapłaty, typ money

--- @id\_faktury - numer identyfikacyjny faktury, typ int

--- @id\_forma\_platnosci - numer identyfikacyjny formy płatności, typ int

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomość wypisana na konsoli informująca o braku faktury o podanym numerze identyfikacyjnym

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_dodaj\_platnosc

@zaplata money,

@id\_faktury int,

@id\_forma\_platnosci int

AS

BEGIN TRY

DECLARE @msg varchar(100)

IF @zaplata IS NULL OR

@id\_faktury IS NULL OR

@id\_forma\_platnosci IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartość NULL'

RAISERROR(16,@msg,1)

END

ELSE

BEGIN

IF((SELECT COUNT(IdFaktury) FROM tbl\_faktura WHERE IdFaktury=@id\_faktury)=1)

BEGIN

INSERT INTO tbl\_platnosc(IdPlatnosci,Zaplata,IdFaktury,IdFormaPlatnosci)

VALUES(NEXT VALUE FOR seq\_new\_id\_platnosci,@zaplata,@id\_faktury,@id\_forma\_platnosci)

PRINT('Pomyślnie dodano rekord')

END

ELSE

BEGIN

PRINT('W bazie nie ma faktury o podanym ID!')

END

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_dodaj\_przeglad\_samochodu(@data\_zrobienia\_przegladu date, @data\_waznosci\_przegladu date, @uwagi varchar(60), @miejsce\_przegladu varchar(30), @cena money, @id\_samochodu int)

--- CREATED BY: Hubert Warchoł

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_przeglad\_samochodu

--- Dodaje informacje na temat przeglądu danego samochodu

---

--- Parametry wejściowe :

--- @data\_zrobienia\_przegladu - data wykonania przeglądu, typ date

--- @data\_waznosci - data wygaśnięcia przeglądu, typ date

--- @uwagi - uwagi do przeglądu, typ varchar

--- @miejsce\_przegladu - miejsce wykonania przeglądu, typ varchar

--- @cena - cena za przegląd, typ money

--- @id\_samochodu - numer identyfikacyjny samochodu, którego tyczy się przegląd, typ int

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomość wypisana na konsoli informująca o braku samochodu o podanym numerze identyfikacyjnym

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_dodaj\_przeglad\_samochodu

@data\_zrobienia\_przegladu date,

@data\_waznosci\_przegladu date,

@uwagi varchar(60),

@miejsce\_przegladu varchar(30),

@cena money,

@id\_samochodu int

AS

BEGIN TRY

DECLARE @msg varchar(100)

IF @data\_zrobienia\_przegladu IS NULL OR

@data\_waznosci\_przegladu IS NULL OR

@miejsce\_przegladu IS NULL OR

@cena IS NULL OR

@id\_samochodu IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartość NULL'

RAISERROR(16,@msg,1)

END

ELSE

BEGIN

IF((SELECT COUNT(IdSamochodu) FROM tbl\_samochod WHERE IdSamochodu = @id\_samochodu)=1)

BEGIN

INSERT INTO tbl\_przeglad\_samochodu(IdPrzegladu,DataZrobieniaPrzegladu,DataWaznosciPrzegladu,Uwagi,MiejsceWykonaniaPrzegladu,Cena,IdSamochodu)

VALUES(NEXT VALUE FOR seq\_new\_id\_przeglad\_samochod,@data\_zrobienia\_przegladu,@data\_waznosci\_przegladu,@uwagi,@miejsce\_przegladu,@cena,@id\_samochodu)

PRINT('Pomyślnie dodano rekord do tabeli')

END

ELSE

BEGIN

PRINT('W bazie nie ma samochodu o podanym ID!')

END

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_dodaj\_rezerwacje(@id\_wypozyczenia int, @id\_samochodu int,@data\_utworzenia\_rezerwacji date)

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_rezerwacja

--- Rejestracja nowej rezerwacji

---

--- Parametry wejściowe :

--- @id\_wypozyczenia, typ int

--- @id\_samochodu - numer identyfikacyjny samochodu, którego tyczy się rezerwacja, typ varchar

--- @data\_utworzenia\_rezerwacji - data złożenia przez klienta rezerwacji, typ date

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_dodaj\_rezerwacje

@id\_wypozyczenia int,

@id\_samochodu int,

@data\_utworzenia\_rezerwacji date

AS

BEGIN TRY

DECLARE @msg varchar(70)

IF @id\_wypozyczenia IS NULL OR

@id\_samochodu IS NULL OR

@data\_utworzenia\_rezerwacji IS NULL

BEGIN

SET @msg = 'Wartości nie mogą przyjmować NULL'

RAISERROR(@msg,16,1)

END

IF (SELECT IdWypozyczenia FROM tbl\_wypozyczenie WHERE IdWypozyczenia = @id\_wypozyczenia) IS NULL

BEGIN

SET @msg = 'Brak wypożyczenia o podanym numerze, dodaj nowe wypożyczenie'

RAISERROR(@msg,16,1)

END

IF (SELECT dbo.uf\_czy\_wolny\_samochod(@id\_samochodu)) = 1

BEGIN

INSERT INTO tbl\_rezerwacja(IdRezerwacji,IdWypozyczenia,IdSamochodu,CenaZaDobe,Kaucja,DataRezerwacji,DataKoncaRezerwacji,DataUtworzeniaRezerwacji,LacznaKwotaDoZaplaty)

VALUES(NEXT VALUE FOR seq\_new\_id\_rezerwacja,@id\_wypozyczenia,@id\_samochodu,dbo.uf\_getCar(@id\_samochodu),1000,dbo.uf\_getDataWypozyczenia(@id\_wypozyczenia),dbo.uf\_getDataZwrotu(@id\_wypozyczenia),@data\_utworzenia\_rezerwacji,(dbo.uf\_czyZwrocicKaucje(@id\_wypozyczenia)+CONVERT(money,dbo.uf\_getCar(@id\_samochodu)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(@id\_wypozyczenia),dbo.uf\_getDataZwrotu(@id\_wypozyczenia)))))

END

ELSE

BEGIN

SET @msg = 'Podany samochód jest niedostępny'

RAISERROR(@msg,16,1)

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_dodaj\_wypozyczenie(@data\_wypozyczenia date, @data\_zwrotu date,@stan\_techniczny varchar(100))

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_wypozyczenie

--- Rejestracja nowego wypożyczenia

---

--- Parametry wejściowe :

--- @data\_wypozyczenia - data wypożyczenia samochodu, typ date

--- @data\_zwrotu - data zwrotu samochodu, typ date

--- @stan\_techniczny - stan techniczny samochodu po zwrocie, typ date

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_dodaj\_wypozyczenie

@data\_wypozyczenia date,

@data\_zwrotu date,

@stan\_techniczny varchar(100)

AS

BEGIN TRY

DECLARE @msg varchar(70)

IF @data\_wypozyczenia IS NULL OR

@data\_zwrotu IS NULL OR

@stan\_techniczny IS NULL

BEGIN

SET @msg = 'Wartości nie mogą przyjmować wartości NULL'

RAISERROR(@msg,16,1)

END

ELSE

BEGIN

INSERT INTO tbl\_wypozyczenie(IdWypozyczenia, DataWypozyczenia, DataZwrotu, StanTechniczny)

VALUES(NEXT VALUE FOR seq\_new\_id\_wypozyczenie,@data\_wypozyczenia,@data\_zwrotu,@stan\_techniczny)

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_dodaj\_ubezpieczenie(@id\_samochodu int, @nr\_polisy varchar(16),@data\_rozp\_ubezpieczenia date, @data\_zakon\_ubezpieczenia date, @firma\_ubezpieczeniowa varchar(50),@cena money)

--- CREATED BY: Hubert Warchoł

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_ubezpieczenie

--- Dodanie nowego ubezpieczenia do bazy danych

---

--- Parametry wejściowe :

--- @id\_samochodu - numer identyfikacyjny samochodu którego tyczy się ubezpieczenie, typ int

--- @nr\_polisy - nr polisy ubezpieczeniowej, typ int

--- @data\_rozp\_ubezpieczenia - data rozpoczęcia czasu trwania ubezpieczenia, typ date

--- @data\_zakon\_ubezpieczenia - data zakończenia trwania ubezpieczenia, typ date

--- @firma\_ubepieczeniowa - nazwa firmy ubezpieczeniowej ,typ varchar

--- @cena - cena za ubezpieczenie, typ money

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_dodaj\_ubezpieczenie

@id\_samochodu int,

@nr\_polisy varchar(16),

@data\_rozp\_ubezpieczenia date,

@data\_zakon\_ubezpieczenia date,

@firma\_ubezpieczeniowa varchar(50),

@cena money

AS

BEGIN TRY

DECLARE @msg varchar(100)

IF @id\_samochodu IS NULL OR

@nr\_polisy IS NULL OR

@data\_rozp\_ubezpieczenia IS NULL OR

@data\_zakon\_ubezpieczenia IS NULL OR

@firma\_ubezpieczeniowa IS NULL OR

@cena IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartość NULL'

RAISERROR(16,@msg,1)

END

ELSE

BEGIN

IF((SELECT COUNT(IdSamochodu) FROM tbl\_samochod WHERE IdSamochodu = @id\_samochodu)=1)

BEGIN

INSERT INTO tbl\_ubezpieczenie(IdUbezpieczenia,IdSamochodu,NumerPolisy,DataRozpoczeciaUbezpieczenia,DataZakonczeniaUbezpieczenia,FirmaUbezpieczeniowa,Cena)

VALUES(NEXT VALUE FOR seq\_new\_id\_ubezpieczenie,@id\_samochodu,@nr\_polisy,@data\_rozp\_ubezpieczenia,@data\_zakon\_ubezpieczenia,@firma\_ubezpieczeniowa,@cena)

PRINT('Pomyślnie dodano rekord do tabeli')

END

ELSE

BEGIN

PRINT('W bazie nie ma samochodu o podanym ID!')

END

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_mod\_fakture(@id\_faktury int,@tytul varchar(50),@wydawca varchar(30),@data\_wystawienia date,@nip varchar(11),@nazwa\_firmy varchar(50),@nr\_konta\_bankowego varchar(28),@id\_rezerwacji int,@id\_wypozyczenia int, @id\_formy\_faktury

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Modyfikuje rekord w tabeli tbl\_faktura

--- Modyfikuje fakture o podanym numerze

---

--- Parametry wejściowe :

--- @id faktury - numer identyfikacyjny faktury, która ma ulec modyfikacji, typ int

--- @tytul - tytuł faktury, typ varchar

--- @wydawca - podmiot wystawiający fakture, typ varchar

--- @data\_wystawienia - data wystawienia faktury, typ date

--- @nip - NIP firmy/osoby na którą wystawiana jest faktura , typ varchar

--- @nazwa\_firmy - nazwa firmy, typ varchar

--- @nr\_konta\_bankowego - numer konta bankowego klienta, typ varchar

--- @id\_rezerwacji - numer identyfikacyjny rezerwacji, typ int

--- @id\_wypozyczenia - numer identyfikacyjny wypożyczenia, typ int

--- @id\_formy faktury - numer identyfikacyjny formy faktury, typ int

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_mod\_fakture

@id\_faktury int,

@tytul varchar(50),

@wydawca varchar(30),

@data\_wystawienia date,

@nip varchar(11),

@nazwa\_firmy varchar(50),

@nr\_konta\_bankowego varchar(28),

@id\_rezerwacji int,

@id\_wypozyczenia int,

@id\_formy\_faktury int

AS

BEGIN TRY

IF((SELECT COUNT(IdFaktury) FROM tbl\_faktura WHERE IdFaktury = @id\_faktury) = 1)

BEGIN

UPDATE tbl\_faktura

SET

Tytyl = @tytul,

Wydawca = @wydawca,

DataWystawienia = @data\_wystawienia,

NIP = @nip,

NazwaFirmy = @nazwa\_firmy,

NrKontaBankowego = @nr\_konta\_bankowego,

IdRezerwacji = @id\_rezerwacji,

IdWypozyczenia = @id\_wypozyczenia,

IdFormyFaktury = @id\_formy\_faktury

WHERE IdFaktury = @id\_faktury

PRINT('Pomyślnie zmieniono rekord w tabeli.')

END

ELSE

BEGIN

PRINT('W bazie nie ma faktury o podanym ID!')

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

GO

CREATE OR ALTER PROCEDURE up\_usun\_fakture

@id\_faktury int

AS

BEGIN TRY

IF((SELECT COUNT(IdFaktury) FROM tbl\_faktura WHERE IdFaktury = @id\_faktury) = 1)

BEGIN

DELETE FROM tbl\_faktura WHERE IdFaktury = @id\_faktury

PRINT('Pomyślnie usunięto rekord w tabeli.')

END

ELSE

BEGIN

PRINT('Upewnij się czy podałeś dobry ID')

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_mod\_klient(@id\_klienta int, @imie varchar(30),@nazwisko varchar(50),@adres varchar(60),@kod\_pocztowy(6),@id\_wojewodztwa int,@id\_kraju int,@telefon varchar(15), @email varchar(40), @data\_urodzenia date, @id\_plec int

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_klient

--- Dodaje nowego klienta do bazy

---

--- Parametry wejściowe :

--- @id\_klienta - numer identyfikacyjny klienta

--- @imie - imie klienta, jest to varchar

--- @nazwisko - nazwisko klienta, jest to varchar

--- @adres - adres klienta, jest to varchar

--- @kod\_pocztowy - kod pocztowy klienta, jest to varchar

--- @id\_województwa - numer identyfikacyjny województwa z którego pochodzi klient, jest to int

--- @id\_kraju - numer identyfikacyjny kraju z którego pochodzi klient, jest to int

--- @telefon - numer telefonu, jest to varchar

--- @email - adres email, jest to varchar

--- @data urodzenia - data urodzenia klienta, jest to date

--- @id\_płci - płeć, jest to int

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_mod\_klient

@id\_klienta int,

@imie varchar(50),

@nazwisko varchar(50),

@adres varchar(60),

@miejscowsc varchar(60),

@kod\_pocztowy varchar(6),

@id\_wojewodztwa int,

@id\_kraj int,

@telefon varchar(15),

@email varchar(40),

@data\_urodzenia date,

@id\_plec int

AS

BEGIN TRY

DECLARE @msg varchar(50)

IF @imie IS NULL OR

@nazwisko IS NULL OR

@adres IS NULL OR

@miejscowsc IS NULL OR

@kod\_pocztowy IS NULL OR

@id\_wojewodztwa IS NULL OR

@id\_kraj IS NULL OR

@telefon IS NULL OR

@data\_urodzenia IS NULL OR

@id\_plec IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartości null'

RAISERROR(@msg,16,1)

END

IF((SELECT COUNT(IdKlienta) FROM tbl\_klient WHERE IdKlienta = @id\_klienta)=1)

BEGIN

UPDATE tbl\_klient

SET

Imie = @imie,

Nazwisko = @nazwisko,

Adres = @adres,

Miejscowosc = @miejscowsc,

KodPocztowy = @kod\_pocztowy,

IdWojewodztwa = @id\_wojewodztwa,

IdKraju = @id\_kraj,

Telefon = @telefon,

Email = @email,

DataUrodzenia = @data\_urodzenia,

IdPlci = @id\_plec

WHERE

IdKlienta = @id\_klienta

PRINT('Pomyślnie zmodyfikowano rekord')

END

ELSE

BEGIN

PRINT('W bazie nie ma klienta o podanym ID')

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_mod\_samochod(@id\_samochodu int, @rejestracja varchar(15),@id\_model int,@id\_marka int,@id\_rodzaj\_nadwozia int,@id\_silnik int,@id\_rodzaj\_samochodu int,@kolor varchar(15), @email varchar(30), @przebieg int, @rocznik int,@vin varchar(17)

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Modyfikuje rekord w tabeli tbl\_samochod

--- Modyfikuje parametry samochodu

---

--- Parametry wejściowe :

--- @rejestracja - rejestracja samochodu, jest to varchar

--- @id\_model - numer identyfikacyjny modelu, jest to int

--- @id\_marka - numer identyfikacyjny marki, jest to int

--- @id\_rodzaj\_nadwozia - numer identyfikacyjny rodzaju nadwozia, jest to int

--- @id\_silnik - numer identyfikacyjny rodzaju silnika, jest to int

--- @id\_rodzaj\_samochodu - - numer identyfikacyjny rodzaju samochodu, jest to int

--- @kolor - kolor samochodu, jest to varchar

--- @przebieg - przebieg samochodu, jest to varchar

--- @rocznik - rocznik samochodu, jest to varchar

--- @vin - numer vin samochodu, jest to varchar

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_mod\_samochod

@id\_samochodu int,

@rejestracja varchar(15),

@id\_model int,

@id\_marka int,

@id\_rodzaj\_nadwozia int,

@id\_silnik int,

@id\_rodzaj\_samochodu int,

@kolor varchar(30),

@przebieg int,

@rocznik int,

@vin varchar(17)

AS

BEGIN TRY

DECLARE @msg varchar(50)

IF @rejestracja IS NULL OR

@id\_model IS NULL OR

@id\_marka IS NULL OR

@id\_rodzaj\_nadwozia IS NULL OR

@id\_silnik IS NULL OR

@id\_rodzaj\_samochodu IS NULL OR

@kolor IS NULL OR

@przebieg IS NULL OR

@rocznik IS NULL OR

@vin IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartości NULL'

RAISERROR(@msg,16,1)

END

ELSE

IF((SELECT COUNT(IdSamochodu) FROM tbl\_samochod WHERE IdSamochodu = @id\_samochodu) = 1)

BEGIN

UPDATE tbl\_samochod

SET

Rejestracja = @rejestracja,

IdModelu = @id\_model,

IdMarki = @id\_marka,

IdRodzajuNadwozia = @id\_rodzaj\_nadwozia,

IdSilnika = @id\_silnik,

IdRodzajuSamochodu = @id\_rodzaj\_samochodu,

Kolor = @kolor,

Przebieg = @przebieg,

Rocznik = @rocznik,

VIN = @vin

WHERE IdSamochodu = @id\_samochodu

PRINT('Pomyślnie zmodyfikowano rekord w tabeli')

END

ELSE

BEGIN

PRINT('W bazie nie ma samochodu o podanym ID!')

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_mod\_cena\_za\_wypozyczenie(@id\_rodzaj\_samochodu int, cena money)

--------------------------------------------------------------------------------

--- Modyfkikuje rekord w tabeli dict.tbl\_rodzaj\_samochodu

--- Zmienia cena danego typu samochodu

---

--- Parametry wejściowe :

--- @id\_rodzaj\_samochodu - numer identyfikacyjny rodzaju samochodu, jest to int

--- @cena - cena za dobe, typ money

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_mod\_cena\_za\_wypozyczenie

@id\_rodzaj\_samochodu int,

@cena money

AS

BEGIN TRY

DECLARE @msg varchar(100)

IF @id\_rodzaj\_samochodu IS NULL OR

@cena IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartości NULL'

RAISERROR(@msg,16,1)

END

ELSE

IF((SELECT COUNT(IdRodzajuSamochodu) FROM dict.tbl\_rodzaj\_samochodu WHERE IdRodzajuSamochodu = @id\_rodzaj\_samochodu)=1)

BEGIN

UPDATE dict.tbl\_rodzaj\_samochodu

SET

Cena = @cena

WHERE IdRodzajuSamochodu = @id\_rodzaj\_samochodu

PRINT('Pomyślnie zaktualizowano cene za dobe')

END

ELSE

BEGIN

PRINT('Upewnij się czy podałeś odpowieni ID!')

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_mod\_ubezpieczenie(@id\_ubezpieczenia int,@id\_samochodu int, @nr\_polisy varchar(16),@data\_rozp\_ubezpieczenia date, @data\_zakon\_ubezpieczenia date, @firma\_ubezpieczeniowa varchar(50),@cena money)

--- CREATED BY: Hubert Warchoł

--------------------------------------------------------------------------------

--- Modyfikuje rekord w tabeli tbl\_ubezpieczenie

--- Zmienia wybrane parametry danego ubezpieczenia

---

--- Parametry wejściowe :

--- @id\_ubezpieczenia - numer identyfikacyjny ubezpieczenia, które ma być poddane modyfikacji

--- @id\_samochodu - numer identyfikacyjny samochodu którego tyczy się ubezpieczenie, typ int

--- @nr\_polisy - nr polisy ubezpieczeniowej, typ int

--- @data\_rozp\_ubezpieczenia - data rozpoczęcia czasu trwania ubezpieczenia, typ date

--- @data\_zakon\_ubezpieczenia - data zakończenia trwania ubezpieczenia, typ date

--- @firma\_ubepieczeniowa - nazwa firmy ubezpieczeniowej ,typ varchar

--- @cena - cena za ubezpieczenie, typ money

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_mod\_ubezpieczenie

@id\_ubezpieczenia int,

@id\_samochodu int,

@nr\_polisy varchar(16),

@data\_rozp\_ubezpieczenia date,

@data\_zakon\_ubezpieczenia date,

@firma\_ubezpieczeniowa varchar(50),

@cena money

AS

BEGIN TRY

DECLARE @msg varchar(100)

IF @id\_ubezpieczenia IS NULL OR

@id\_samochodu IS NULL OR

@nr\_polisy IS NULL OR

@data\_rozp\_ubezpieczenia IS NULL OR

@data\_zakon\_ubezpieczenia IS NULL OR

@firma\_ubezpieczeniowa IS NULL OR

@cena IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartość NULL'

RAISERROR(16,@msg,1)

END

ELSE

BEGIN

IF((SELECT COUNT(IdUbezpieczenia) FROM tbl\_ubezpieczenie WHERE IdUbezpieczenia=@id\_ubezpieczenia)=1)

BEGIN

IF((SELECT COUNT(IdSamochodu) FROM tbl\_samochod WHERE IdSamochodu = @id\_samochodu)=1)

BEGIN

UPDATE tbl\_ubezpieczenie

SET

IdUbezpieczenia = @id\_ubezpieczenia,

IdSamochodu = @id\_samochodu,

NumerPolisy = @nr\_polisy,

DataRozpoczeciaUbezpieczenia = @data\_rozp\_ubezpieczenia,

DataZakonczeniaUbezpieczenia = @data\_zakon\_ubezpieczenia,

FirmaUbezpieczeniowa = @firma\_ubezpieczeniowa,

Cena = @cena

WHERE IdUbezpieczenia = @id\_ubezpieczenia

PRINT('Pomyślnie zmodyfikowano rekord w tabeli')

END

ELSE

BEGIN

PRINT('W bazie nie ma samochodu o podanym ID!')

END

END

ELSE

BEGIN

PRINT('W bazie nie ma ubezpieczenia o podanym ID!')

END

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_usun\_ubezpieczenie(@id\_ubezpieczenia int)

--- CREATED BY: Hubert Warchoł

--------------------------------------------------------------------------------

--- Usuwa rekord w tabeli tbl\_ubezpieczenie

--- Usuwa dane ubezpieczenie

---

--- Parametry wejściowe :

--- @id\_ubezpieczenia - numer identyfikacyjny ubezpieczenia, które ma być usunięte

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_usun\_ubezpieczenie

@id\_ubezpieczenia int

AS

BEGIN TRY

DECLARE @msg varchar(100)

IF @id\_ubezpieczenia IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartość NULL'

RAISERROR(16,@msg,1)

END

ELSE

BEGIN

IF((SELECT COUNT(IdUbezpieczenia) FROM tbl\_ubezpieczenie WHERE IdUbezpieczenia=@id\_ubezpieczenia)=1)

BEGIN

DELETE FROM tbl\_ubezpieczenie WHERE IdUbezpieczenia = @id\_ubezpieczenia

PRINT('Pomyślnie usunięto rekord z bazy')

END

ELSE

BEGIN

PRINT('W bazie nie ma ubezpieczenia o podanym ID!')

END

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_mod\_rezerwacja(@id\_rezerwacji int,@id\_wypozyczenia int, @id\_samochodu int,@data\_utworzenia\_rezerwacji date)

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Modyfikuje rekord w tabeli tbl\_rezerwacja

--- Modyfikuje parametry danej rezerwacji

---

--- Parametry wejściowe :

--- @id\_rezerwacji - numer identyfikacyjny rezerwacji, która ma być zmodyfikowana, typ int

--- @id\_wypozyczenia, typ int

--- @id\_samochodu - numer identyfikacyjny samochodu, którego tyczy się rezerwacja, typ varchar

--- @data\_utworzenia\_rezerwacji - data złożenia przez klienta rezerwacji, typ date

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_mod\_rezerwacja

@id\_rezerwacji int,

@id\_wypozyczenia int,

@id\_samochodu int,

@data\_utworzenia\_rezerwacji date

AS

BEGIN TRY

DECLARE @msg varchar(70)

IF @id\_wypozyczenia IS NULL OR

@id\_samochodu IS NULL OR

@data\_utworzenia\_rezerwacji IS NULL

BEGIN

SET @msg = 'Wartości nie mogą przyjmować NULL'

RAISERROR(@msg,16,1)

END

IF ((SELECT COUNT(IdRezerwacji) FROM tbl\_rezerwacja WHERE IdRezerwacji = @id\_rezerwacji)=1)

BEGIN

IF((SELECT COUNT(IdWypozyczenia) FROM tbl\_wypozyczenie WHERE IdWypozyczenia = @id\_wypozyczenia)=1)

BEGIN

IF((SELECT COUNT(IdSamochodu) FROM tbl\_samochod WHERE IdSamochodu = @id\_samochodu)=1)

BEGIN

UPDATE tbl\_rezerwacja

SET

IdWypozyczenia = @id\_wypozyczenia,

IdSamochodu = @id\_samochodu,

CenaZaDobe = dbo.uf\_getCar(@id\_samochodu),

DataRezerwacji = dbo.uf\_getDataWypozyczenia(@id\_wypozyczenia),

DataKoncaRezerwacji = dbo.uf\_getDataZwrotu(@id\_wypozyczenia),

DataUtworzeniaRezerwacji = @data\_utworzenia\_rezerwacji,

LacznaKwotaDoZaplaty = dbo.uf\_czyZwrocicKaucje(@id\_wypozyczenia)+CONVERT(money,dbo.uf\_getCar(@id\_samochodu)\*DATEDIFF(day,dbo.uf\_getDataWypozyczenia(@id\_wypozyczenia),dbo.uf\_getDataZwrotu(@id\_wypozyczenia)))

WHERE IdRezerwacji = @id\_rezerwacji

PRINT('Pomyślnie zmodyfikowano rekord')

END

ELSE

BEGIN

PRINT('W bazie nie ma samochodu o podanym ID!')

END

END

ELSE

BEGIN

PRINT('W bazie nie ma wypożyczenia o podanym ID!')

END

END

ELSE

BEGIN

PRINT('W bazie nie ma rezerwacji o podanym ID')

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_mod\_wypozyczenie(@id\_wypozyczenia int,@data\_wypozyczenia date, @data\_zwrotu date,@stan\_techniczny varchar(100))

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Modyfikuje rekord w tabeli tbl\_wypozyczenie

--- Modyfikuje parametry danego wypożyczenia

---

--- Parametry wejściowe :

--- @id\_wypozyczenia - numer identyfikacyjny wypożyczenia, które ma być zmodyfikowane, typ int

--- @data\_wypozyczenia - data wypożyczenia samochodu, typ date

--- @data\_zwrotu - data zwrotu samochodu, typ date

--- @stan\_techniczny - stan techniczny samochodu po zwrocie, typ date

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_mod\_wypozyczenie

@id\_wypozyczenia int,

@data\_wypozyczenia date,

@data\_zwrotu date,

@stan\_techniczny varchar(100)

AS

BEGIN TRY

DECLARE @msg varchar(70)

IF @id\_wypozyczenia IS NULL OR

@data\_wypozyczenia IS NULL OR

@data\_zwrotu IS NULL OR

@stan\_techniczny IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartość NULL'

RAISERROR(16,@msg,1)

END

ELSE

BEGIN

IF((SELECT COUNT(IdWypozyczenia) FROM tbl\_wypozyczenie WHERE IdWypozyczenia = @id\_wypozyczenia)=1)

BEGIN

UPDATE tbl\_wypozyczenie

SET

DataWypozyczenia = @data\_wypozyczenia,

DataZwrotu = @data\_zwrotu,

StanTechniczny = @stan\_techniczny

WHERE IdWypozyczenia = @id\_wypozyczenia

END

ELSE

BEGIN

PRINT('W bazie nie ma wypożyczenia o podanym ID!')

END

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_usun\_fakture(@id\_faktury int)

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Usuwa rekord w tabeli tbl\_faktura

--- Usuwa fakture o podanym numerze

---

--- Parametry wejściowe :

--- @id faktury - numer identyfikacyjny faktury, która ma być usunięta, typ int

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_usun\_fakture

@id\_faktury int

AS

BEGIN TRY

IF((SELECT COUNT(IdFaktury) FROM tbl\_faktura WHERE IdFaktury = @id\_faktury) = 1)

BEGIN

DELETE FROM tbl\_faktura WHERE IdFaktury = @id\_faktury

PRINT('Pomyślnie usunięto rekord w tabeli.')

END

ELSE

BEGIN

PRINT('Upewnij się czy podałeś dobry ID')

END

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

DECLARE @ErrorSeverity INT;

DECLARE @ErrorState INT;

SELECT

@ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE();

RAISERROR(@ErrorMessage,@ErrorSeverity,@ErrorState)

END CATCH

--------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_powiazanie\_klienta\_z\_rezerwacja(@id\_klienta int,@id\_rezerwacji)

--- CREATED BY: Igor Owczarek

--------------------------------------------------------------------------------

--- Dodaje rekord do tabeli tbl\_rezerwacja\_klient

--- Dokonuje powiązania danego klienta z konkretną rezerwacją

---

--- Parametry wejściowe :

--- @id\_klienta - numer identyfikacyjny klient, typ int

--- @id\_rezerwacji - numer identyfikacyjny rezerwacji, typ int

---

--- Parametry wyjściowe :

--- Wiadomość wypisana na konsoli informująca o poporawnym wykonaniu procedury

--- Wiadomośc wypisana na konsoli informująca o błędnym wykonaniu procedury

--------------------------------------------------------------------------------

CREATE OR ALTER PROCEDURE up\_powiazanie\_klienta\_z\_rezerwacja

@id\_klienta int,

@id\_rezerwacji int

AS

BEGIN TRY

DECLARE @msg varchar(50)

IF @id\_klienta IS NULL OR

@id\_rezerwacji IS NULL

BEGIN

SET @msg = 'Kolumny nie mogą przyjmować wartości null'

RAISERROR(@msg,16,1)

END

ELSE

BEGIN

IF ((SELECT COUNT(IdRezerwacji) FROM tbl\_rezerwacja WHERE IdRezerwacji = @id\_rezerwacji) = 1)

BEGIN

IF((SELECT COUNT(IdKlienta) FROM tbl\_klient WHERE IdKlienta = @id\_klienta) = 1)

BEGIN

INSERT INTO tbl\_rezerwacja\_klient

VALUES(@id\_klienta,@id\_rezerwacji)

END

ELSE

BEGIN

PRINT('W bazie nie ma klienta o podanym ID!')

END

END

ELSE

BEGIN

PRINT('W bazie nie ma rezerwacji o podanym ID!')

END

END

END TRY

BEGIN CATCH

SELECT

[ErrorLine] = ERROR\_LINE(),

[ErrorNumber] = ERROR\_NUMBER(),

[ErrorSeverity] = ERROR\_SEVERITY(),

[ErrorState] = ERROR\_STATE(),

[ErrorMessage] = ERROR\_MESSAGE()

END CATCH

# Wykaz utworzonych funkcji użytkownika

Obraz zawierający tekst

Opis wygenerowany automatycznie

# Skrypt tworzenia funkcji definiowanych przez użytkownika

CREATE FUNCTION uf\_getCar(@id\_samochodu int)

RETURNS money

BEGIN

DECLARE @cena money

SET @cena = (SELECT Cena FROM dict.tbl\_rodzaj\_samochodu c INNER JOIN tbl\_samochod s ON s.IdRodzajuSamochodu = c.IdRodzajuSamochodu WHERE s.IdSamochodu = @id\_samochodu)

RETURN @cena

END

GO

CREATE FUNCTION uf\_getDataWypozyczenia(@id\_wypozyczenia int)

RETURNS date

BEGIN

DECLARE @data\_wypozyczenia date

SET @data\_wypozyczenia = (SELECT DataWypozyczenia FROM tbl\_wypozyczenie WHERE IdWypozyczenia = @id\_wypozyczenia)

RETURN @data\_wypozyczenia

END

GO

CREATE FUNCTION uf\_getDataZwrotu(@id\_wypozyczenia int)

RETURNS date

BEGIN

DECLARE @data\_zwrotu date

SET @data\_zwrotu = (SELECT DataZwrotu FROM tbl\_wypozyczenie WHERE IdWypozyczenia = @id\_wypozyczenia)

RETURN @data\_zwrotu

END

GO

CREATE OR ALTER FUNCTION uf\_czyZwrocicKaucje(@id\_wypozyczenia int)

RETURNS money

BEGIN

DECLARE @kaucja money

IF (SELECT StanTechniczny FROM tbl\_wypozyczenie WHERE IdWypozyczenia = @id\_wypozyczenia) = 'dobry'

SET @kaucja = 0

ELSE

SET @kaucja = 1000

RETURN @kaucja

END

GO

CREATE OR ALTER FUNCTION uf\_getCarId(@id\_faktury int)

RETURNS int

BEGIN

DECLARE @id int

SET @id = (SELECT IdSamochodu FROM tbl\_rezerwacja r INNER JOIN tbl\_faktura f ON r.IdRezerwacji = f.IdRezerwacji WHERE IdFaktury=@id\_faktury)

RETURN @id

END

GO

CREATE OR ALTER FUNCTION uf\_getCarType(@id\_faktury int)

RETURNS tinyint

BEGIN

DECLARE @id int

SET @id = (SELECT IdRodzajuSamochodu FROM tbl\_samochod s INNER JOIN tbl\_rezerwacja r ON r.IdSamochodu = s.IdSamochodu INNER JOIN tbl\_faktura f ON f.IdRezerwacji = r.IdRezerwacji WHERE IdFaktury = @id\_faktury)

RETURN @id

END

GO

CREATE OR ALTER FUNCTION uf\_getLacznaKwota(@id\_faktury int)

RETURNS money

BEGIN

DECLARE @laczna\_kwota money

SET @laczna\_kwota = (SELECT LacznaKwotaDoZaplaty FROM tbl\_rezerwacja r INNER JOIN tbl\_faktura f ON r.IdRezerwacji = f.IdRezerwacji WHERE IdFaktury=@id\_faktury)

RETURN @laczna\_kwota

END

GO

CREATE OR ALTER FUNCTION uf\_czy\_wolny\_samochod(@id\_samochodu int)

RETURNS bit

BEGIN

DECLARE @czy\_wolny bit

IF (SELECT DataZwrotu FROM tbl\_wypozyczenie w INNER JOIN tbl\_rezerwacja r ON r.IdWypozyczenia = w.IdWypozyczenia WHERE IdSamochodu=@id\_samochodu AND DataZwrotu>=GETDATE())>=GETDATE()

SET @czy\_wolny =0

ELSE

SET @czy\_wolny=1

RETURN @czy\_wolny

END

CREATE FUNCTION uf\_wyszukaj\_marke\_po\_nazwie

(@marka varchar(30))

RETURNS TABLE

AS

RETURN(

SELECT \* FROM dict.tbl\_marka

WHERE LOWER(Marka) LIKE LOWER('%'+@marka+'%'))

GO

CREATE FUNCTION uf\_wyszukaj\_model\_po\_nazwie

(@model varchar(50))

RETURNS TABLE

AS

RETURN(

SELECT \* FROM dict.tbl\_model

WHERE LOWER(Model) LIKE LOWER('%'+@model+'%'))

GO

CREATE FUNCTION uf\_wyszukaj\_klienta\_po\_imieniu\_nazwisku

(@imieLubNazwisko varchar(60))

RETURNS TABLE

AS

RETURN(

SELECT \* FROM tbl\_klient

WHERE LOWER(Imie)+' '+LOWER(Nazwisko) LIKE LOWER('%'+@imieLubNazwisko+'%'))

GO

CREATE OR ALTER FUNCTION uf\_wyszukaj\_samochod\_po\_marce

(@model varchar(50))

RETURNS TABLE

AS

RETURN(

SELECT s.IdSamochodu,s.Rejestracja,m.Marka,model.Model FROM tbl\_samochod s INNER JOIN dict.tbl\_marka m ON s.IdMarki = m.IdMarki

INNER JOIN dict.tbl\_model model ON model.IdModelu = s.IdModelu

WHERE LOWER(m.Marka) LIKE LOWER('%'+@model+'%'))

GO

CREATE OR ALTER FUNCTION uf\_ile\_rezerwacji\_ma\_klient

(@imieLubNazwisko varchar(50))

RETURNS TABLE

AS

RETURN(

SELECT k.Imie,k.Nazwisko, COUNT(rk.IdKlienta) AS 'Liczba rezerwacji klienta' FROM tbl\_rezerwacja\_klient rk INNER JOIN tbl\_klient k ON rk.IdKlienta = k.IdKlienta

WHERE LOWER(k.Imie) + ' '+LOWER(k.Nazwisko) LIKE LOWER('%'+@imieLubNazwisko+'%')

GROUP BY rk.IdKlienta,k.Imie,k.Nazwisko)

GO

CREATE OR ALTER FUNCTION uf\_wyswietl\_dane\_z\_kursora\_rezerwacje()

RETURNS @kursor TABLE(wartosc varchar(255))

AS

BEGIN

DECLARE @imie varchar(50);

DECLARE @nazwisko varchar(50);

DECLARE @model varchar(50);

DECLARE @marka varchar(50);

DECLARE crs\_rezerwacje\_klienta CURSOR FOR

SELECT k.Imie, k.Nazwisko, m.Marka,model.Model FROM tbl\_klient k

INNER JOIN tbl\_rezerwacja\_klient rk ON rk.IdKlienta = k.IdKlienta

INNER JOIN tbl\_rezerwacja r ON r.IdRezerwacji = rk.IdRezerwacji

INNER JOIN tbl\_samochod s ON r.IdSamochodu = s.IdSamochodu

INNER JOIN dict.tbl\_marka m ON m.IdMarki = s.IdMarki

INNER JOIN dict.tbl\_model model ON model.IdModelu = s.IdSamochodu;

OPEN crs\_rezerwacje\_klienta;

FETCH NEXT FROM crs\_rezerwacje\_klienta INTO @imie,@nazwisko,@marka,@model;

WHILE @@FETCH\_STATUS=0

BEGIN

INSERT INTO @kursor(wartosc) VALUES('Klient '+@imie+' '+@nazwisko+' wypożyczył\a '+@marka+' '+@model);

FETCH NEXT FROM crs\_rezerwacje\_klienta INTO @imie,@nazwisko,@marka,@model;

END

CLOSE crs\_rezerwacje\_klienta;

DEALLOCATE crs\_rezerwacje\_klienta;

RETURN

END

GO

CREATE OR ALTER FUNCTION uf\_wyswietl\_dane\_z\_kursora\_faktury()

RETURNS @kursor TABLE( wartosc varchar(255))

AS

BEGIN

DECLARE @id int;

DECLARE @tytul varchar(50);

DECLARE @calkowita\_wartosc int;

DECLARE crs\_wartosc\_faktury CURSOR FOR

SELECT f.IdFaktury,f.Tytyl,SUM(pf.Cena) FROM tbl\_faktura f INNER JOIN tbl\_pozycje\_faktury pf ON f.IdFaktury = pf.IdFaktury

GROUP BY f.IdFaktury,f.Tytyl;

OPEN crs\_wartosc\_faktury;

FETCH NEXT FROM crs\_wartosc\_faktury into @id,@tytul,@calkowita\_wartosc;

WHILE @@FETCH\_STATUS = 0

BEGIN

INSERT INTO @kursor(wartosc) VALUES('Całkowita wartość faktury '+@tytul+' wynosi '+CONVERT(varchar(50),@calkowita\_wartosc)+' zł')

FETCH NEXT FROM crs\_wartosc\_faktury into @id,@tytul,@calkowita\_wartosc;

END

CLOSE crs\_wartosc\_faktury;

DEALLOCATE crs\_wartosc\_faktury;

RETURN

END

GO

CREATE FUNCTION uf\_wyszukaj\_fakture\_po\_nr

(@nr\_faktury int)

RETURNS TABLE

AS

RETURN(

SELECT \* FROM tbl\_faktura

WHERE IdFaktury = @nr\_faktury)

GO

CREATE FUNCTION uf\_wyszukaj\_wojewodztwo\_po\_nazwie

(@wojewodztwo varchar(25))

RETURNS TABLE

AS

RETURN(

SELECT IdWojewodztwa,Wojewodztwo FROM dict.tbl\_wojewodztwo

WHERE LOWER(Wojewodztwo) LIKE LOWER('%'+@wojewodztwo+'%'))

go

CREATE FUNCTION uf\_wyszukaj\_ubezpieczenie\_po\_nr\_polisy

(@nr\_polisy varchar(16))

RETURNS TABLE

AS

RETURN(

SELECT \* FROM tbl\_ubezpieczenie

WHERE LOWER(NumerPolisy) LIKE LOWER('%'+@nr\_polisy+'%'))

GO

CREATE FUNCTION uf\_wyszukaj\_platnosci\_po\_tytule\_faktury

(@tytul varchar(50))

RETURNS TABLE

AS

RETURN(

SELECT p.IdPlatnosci, p.IdFaktury,f.Tytyl,p.Zaplata,p.IdFormaPlatnosci FROM tbl\_platnosc p

INNER JOIN tbl\_faktura f

ON p.IdFaktury = f.IdFaktury

WHERE LOWER(f.Tytyl) LIKE LOWER('%'+@tytul+'%'))

GO

CREATE FUNCTION uf\_wyszukaj\_klienta\_po\_wojewodztwie

(@wojewodztwo varchar(19))

RETURNS TABLE

AS

RETURN(

SELECT IdKlienta,Imie,Nazwisko,Adres,Miejscowosc,KodPocztowy,w.Wojewodztwo,IdKraju,Telefon,Email,DataUrodzenia,IdPlci FROM tbl\_klient k

INNER JOIN dict.tbl\_wojewodztwo w

ON k.IdWojewodztwa = w.IdWojewodztwa

WHERE LOWER(w.Wojewodztwo) LIKE LOWER('%'+@wojewodztwo+'%'))

GO

CREATE OR ALTER FUNCTION uf\_ile\_klientek()

RETURNS TABLE

AS

RETURN(

SELECT COUNT(IdKlienta) AS 'Liczba klientek' FROM tbl\_klient WHERE IdPlci = 1 GROUP BY IdPlci);

GO

CREATE OR ALTER FUNCTION uf\_ile\_klientow()

RETURNS TABLE

AS

RETURN(

SELECT COUNT(IdKlienta) AS 'Liczba klientow' FROM tbl\_klient WHERE IdPlci = 2 GROUP BY IdPlci);

GO

CREATE OR ALTER FUNCTION uf\_wyswietl\_dane\_z\_kursora\_klienta()

RETURNS @kursor TABLE(wartosc varchar(255))

AS

BEGIN

DECLARE @imie varchar(30);

DECLARE @nazwisko varchar(50);

DECLARE @telefon varchar(15);

DECLARE @email varchar(40);

DECLARE crs\_kontakt\_klienci CURSOR FOR

SELECT Imie,Nazwisko,Telefon,Email FROM tbl\_klient

OPEN crs\_kontakt\_klienci;

FETCH NEXT FROM crs\_kontakt\_klienci INTO @imie,@nazwisko,@telefon,@email;

WHILE @@FETCH\_STATUS = 0

BEGIN

INSERT INTO @kursor(wartosc) VALUES('Kontakt do klienta: '+@imie+' '+@nazwisko+' | '+@telefon+' | '+@email)

FETCH NEXT FROM crs\_kontakt\_klienci INTO @imie,@nazwisko,@telefon,@email;

END

CLOSE crs\_kontakt\_klienci;

DEALLOCATE crs\_kontakt\_klienci;

RETURN

END

GO

CREATE FUNCTION uf\_wyszukaj\_samochod\_po\_vinie

(@vin varchar(17))

RETURNS TABLE

AS

RETURN(

SELECT \* FROM tbl\_samochod

WHERE LOWER(VIN) LIKE LOWER('%'+@vin+'%'))

GO

CREATE FUNCTION uf\_wyszukaj\_klienta\_po\_nipie

(@nip varchar(11))

RETURNS TABLE

AS

RETURN(

SELECT k.IdKlienta,k.Imie,k.Nazwisko,k.Miejscowosc,k.Adres,k.Telefon,f.NIP FROM tbl\_klient k

INNER JOIN tbl\_rezerwacja\_klient rk ON rk.IdKlienta = k.IdKlienta

JOIN tbl\_rezerwacja r ON rk.IdRezerwacji = r.IdRezerwacji

JOIN tbl\_faktura f ON r.IdRezerwacji = f.IdRezerwacji

WHERE LOWER(f.NIP) LIKE LOWER('%'+@nip+'%'))

# Wykaz utworzonych triggerów











# Skrypt z tworzenia triggerów

-------------------------------------------------------------------------------------

--- TRIGGER DEFINITION

--- trgtbl\_klient

--- CREATE TRIGGER trgtbl\_klient ON tbl\_klient FOR DELETE

--- CREATED BY: Hubert Warchoł

-------------------------------------------------------------------------------------

--- Uniemożliwia usunięcie klienta z tabeli tbl\_klient

--- Wykonuje się dla operacji DELETE

--- Parametry wyjściowe: Informacja o braku możliwości usunięcia klienta

-------------------------------------------------------------------------------------

CREATE TRIGGER trgtbl\_klient ON tbl\_klient

FOR DELETE

AS SET NOCOUNT ON

BEGIN

ROLLBACK TRANSACTION

PRINT('Nie można usuwac klientów z bazy!')

END

-------------------------------------------------------------------------------------

--- TRIGGER DEFINITION

--- trgtbl\_klient\_kod

--- CREATE TRIGGER trgtbl\_klient\_kod ON tbl\_klient FOR INSERT, UPDATE

--- CREATED BY: Hubert Warchoł

-------------------------------------------------------------------------------------

--- Tworzy kod dla klienta z jego id, imienia i nazwiska

--- Wykonuje się dla operacji INSERT i UPDATE

--- Parametry wyjściowe: Wprowadza kod klienta do danej kolumny

-------------------------------------------------------------------------------------

CREATE TRIGGER trgtbl\_klient\_kod ON tbl\_klient

FOR INSERT, UPDATE

AS

UPDATE tbl\_klient

SET KodKlienta=(SELECT REPLICATE ('0',(4-(DATALENGTH(CONVERT(varchar(15),i.IdKlienta))))) + CONVERT(varchar(10),i.IdKlienta)+SUBSTRING(i.Nazwisko,1,3)+SUBSTRING(i.Imie,1,1)

FROM tbl\_klient k INNER JOIN inserted i ON i.IdKlienta = k.IdKlienta) FROM tbl\_klient k INNER JOIN inserted i ON i.IdKlienta = k.IdKlienta

-------------------------------------------------------------------------------------

--- TRIGGER DEFINITION

--- trgtbl\_faktura

--- CREATE TRIGGER trgtbl\_faktura ON tbl\_faktura FOR DELETE

--- CREATED BY: Hubert Warchoł

-------------------------------------------------------------------------------------

--- Uniemożliwia usunięcie faktury jeżeli jest rok wystawienia zgadza się z aktualnym rokiem

--- Wykonuje się dla operacji DELETE

--- Parametry wyjściowe: Informacja w zależności od spełnienia warunku – dla true informacja o tym że nie można usunąć, a dla false, że pomyślnie usunięto.

-------------------------------------------------------------------------------------

CREATE TRIGGER trgtbl\_faktura ON tbl\_faktura

FOR DELETE

AS SET NOCOUNT ON

BEGIN

DECLARE @currentYear int = CONVERT(int, YEAR(GETDATE()))

DECLARE @yearOfInvoice int = CONVERT(int, YEAR((SELECT DataWystawienia FROM deleted)))

IF(@yearOfInvoice = @currentYear)

BEGIN

PRINT('Nie można usunąć rezerwacji')

ROLLBACK TRANSACTION

END

ELSE

PRINT('Pomyślnie usunięto rezerwację z bazy')

END

-------------------------------------------------------------------------------------

--- TRIGGER DEFINITION

--- trgtbl\_wojewodztwa

--- CREATE TRIGGER trgtbl\_wojewodztwa ON dict.tbl\_wojewodztwo FOR INSERT

--- CREATED BY: Hubert Warchoł

-------------------------------------------------------------------------------------

--- Uniemożliwia dodanie kolejnego województwa

--- Wykonuje się dla operacji INSERT

--- Parametry wyjściowe: Błąd informujący o tym, że nie można dodać województwa

-------------------------------------------------------------------------------------

CREATE TRIGGER trgtbl\_wojewodztwa ON dict.tbl\_wojewodztwo

FOR INSERT

AS

DECLARE @ile int

SET @ile=(SELECT COUNT(\*) FROM dict.tbl\_wojewodztwo)

IF @ile>16

BEGIN

RAISERROR ('Nie możesz dodać kolejnego województwa',10,1)

ROLLBACK TRANSACTION

END

CREATE TRIGGER trgsrv\_tworzenie\_bd

ON ALL SERVER

AFTER CREATE\_DATABASE

AS

BEGIN

PRINT 'Stworzono baze danych'

END

CREATE TRIGGER trgdb\_drop\_i\_alter

ON DATABASE

FOR DROP\_TABLE, ALTER\_TABLE

AS

BEGIN

PRINT 'Nie można wykonać zapytań DROP i ALTER'

ROLLBACK;

END

# Wykaz obiektów wartości domyślnych oraz reguł

Obraz zawierający tekst

Opis wygenerowany automatycznie

CREATE DEFAULT df\_data\_wystawienia AS CAST(GETDATE() AS DATE)

EXEC sp\_bindefault df\_data\_wystawienia, 'tbl\_faktura.DataWystawienia';

CREATE DEFAULT df\_data\_rezerwacji AS GETDATE()

EXEC sp\_bindefault df\_data\_rezerwacji, 'tbl\_rezerwacja.DataRezerwacji'

CREATE DEFAULT df\_data\_utworzenia\_rezerwacji AS GETDATE();

EXEC sp\_bindefault df\_data\_rezerwacji, 'tbl\_rezerwacja.DataUtworzeniaRezerwacji'

CREATE DEFAULT df\_data\_wypozyczenia AS GETDATE()

EXEC sp\_bindefault df\_data\_wypozyczenia, 'tbl\_wypozyczenie.DataWypozyczenia'

CREATE DEFAULT df\_telefon AS '+000-000-000-00'

EXEC sp\_bindefault df\_telefon, 'tbl\_klient.Telefon'

CREATE DEFAULT df\_email AS 'default.mail@default.com'

EXEC sp\_bindefault df\_email, 'tbl\_klient.Email'

# Wykaz obiektów reguł

Obraz zawierający tekst

Opis wygenerowany automatycznie

CREATE RULE rl\_kod\_pocztowy as @kodPocztowy LIKE '[0-9][0-9]-[0-9][0-9][0-9]';

EXEC sp\_bindrule rl\_kod\_pocztowy, 'tbl\_klient.KodPocztowy';

CREATE RULE rl\_email as @email LIKE '%[A-Z0-9][@][A-Z0-9]%[.][A-Z0-9]%'

OR @email LIKE NULL;

EXEC sp\_bindrule rl\_email, 'tbl\_klient.Email'

CREATE RULE rl\_data\_urodzenia AS (@dataUrodzenia < CAST(GETDATE() AS DATE));

EXEC sp\_bindrule rl\_data\_urodzenia, 'tbl\_klient.DataUrodzenia'

CREATE RULE rl\_data\_wystawienia\_fk AS (@dataWystawieniaFk <= CAST(GETDATE() AS DATE));

EXEC sp\_bindrule rl\_data\_wystawienia\_fk, 'tbl\_faktura.DataWystawienia'

CREATE RULE rl\_data\_utworzenia\_rezerwacji AS (@dataUtwRezerwacji <= CAST(GETDATE() AS DATE));

EXEC sp\_bindrule rl\_data\_utworzenia\_rezerwacji,'tbl\_rezerwacja.DataUtworzeniaRezerwacji'

CREATE RULE rl\_data\_rezerwacji AS (@dataRezerwacji <= CAST(GETDATE() AS DATE));

EXEC sp\_bindrule rl\_data\_rezerwacji, 'tbl\_rezerwacja.DataRezerwacji'

# Wnioski i podsumowanie

Baza danych jest zaprojektowana według zasad projektowania baz danych. Spełnia wyżej wymienione wymagania i funkcjonalności. Została wypełniona przykładowymi danymi.

Baza posiada wiele widoków, które ułatwiają analizę danych klientów, rezerwacji czy samochodów. Dzięki niej również można dodawać i modyfikować dane dotyczące klientów, rezerwacji, samochodów, czy ubezpieczeń. Posiada wiele funkcji ułatwiających pracę z danymi. Pozwala na zarządzanie fakturami i płatnościami.

Projekt został stworzony z myślą o każdej operacji prowadzonej w wypożyczalni samochodów, tak aby był on w pełni funkcjonalny dla osób ją prowadzących.