

Projekt Modelu Bazy Danych

DZIAŁALNOŚĆ WARSZTATU SAMOCHODOWEGO

1. Opis Problemu

Warsztat samochodowy zamówił projekt bazy danych, mającej usprawnić jego działalność, w zakresie wykonywanych napraw samochodów. W bazie należy przechować dane:

- klienta (imię, nazwisko, adres zamieszkania, pesel),
- oddawanego do naprawy samochodu (marka, model, rodzaj silnika, rejestracja),
- informacje o wykonanej naprawie (kwota zapłaty, data oddania samochodu do naprawy, data odbioru samochodu) – na naprawę może składać się wiele różnych czynności (nazwa, opis, czas wykonania, koszt),
- pracownika (imię, nazwisko, pesel, adres zamieszkania, data zatrudnienia)

Ważne informacje od klienta:

Każda naprawa musi mieć przydzielonego pracownika, który jest za nią odpowiedzialny.

Czynności w naprawie, mogą być wykonywane przez różnych pracowników, oraz więcej niż jednego pracownika.

Informacje o modelu i marce mają być danymi dynamicznymi.

2. Model Danych

3.1 Słownik Pojęć

- **Klient** = osoba, która zleca usługę, w zamian oddając konkretną płatność, adekwatną do wykonywanej usługi.
- **PESEL** = jedenastocyfrowy symbol numeryczny, który jednoznacznie identyfikuje osobę fizyczną.
- **Marka Samochodu** = znak firmowy producenta samochodu, zazwyczaj związany z jego nazwą.
- **Model Samochodu** = konkretna nazwa, nadawana pojazdowi o konkretnych specyfikacjach, w celu odróżnienia go od reszty samochodów tej samej marki.

- **Rodzaj Silnika** = kodowane opisanie rodzaju silnika, kod można znaleźć na bloku silnika lub jeśli jest on fabryczny, przez stronę producenta.
- **Rejestracja** = kod literowo-cyfrowy umożliwiający identyfikację każdego pojazdu. Kody mogą mieć różną długość, co często uwarunkowane jest krajem produkcji.
- **Pracownik** = osoba zatrudniona w firmie, lub przez pojedynczą jednostkę, posiadająca konkretne doświadczenie potrzebne do wykonywanej pracy. W zamian za świadczenie konkretnych usług, otrzymuje wcześniej ustalone wynagrodzenie.
- **Kwota Zapłaty** = suma pieniędzy, którą klient musi zapłacić w ramach wykonania wszystkich zleconych usług.
- **Data Zatrudnienia** = dokładna data podpisania umowy o pracę w danym miejscu, przez wskazaną jednostkę.

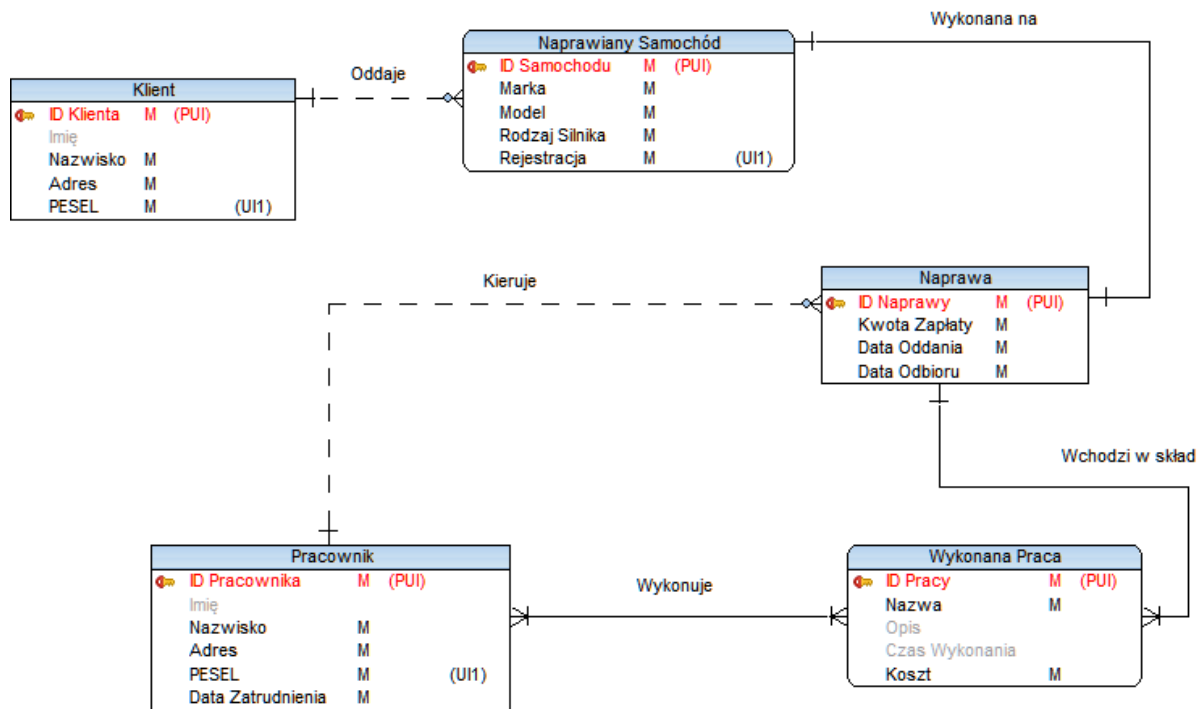
3.2 Fakty (reguły biznesowe)

Klient rejestrowany jest w bazie danych w momencie zostawienia samochodu w warsztacie i zlecenia naprawy. Po wykonaniu przybliżonej wyceny, klient zostaje poinformowany o przewidywanym czasie odbioru auta, oraz zapytany o zgodę w sprawie przybliżonego kosztu naprawy. Jeśli zlecenie jest nie konkretne i klient życzy sobie znaleźć usterkę, wycena zostanie przedstawiona po wykonaniu diagnozy. W przypadku braku zgody na poniesienie kosztów całej lub części naprawy, klient zmuszony jest ponieść koszty diagnozy. Nie można otrzymać zwrotu za wykonaną już naprawę.

Kwota zapłaty to suma kosztów wszystkich wykonanych prac wchodzących w skład naprawy.

- 1 Klient może zlecić naprawę wielu samochodów.
- 2 Samochód może mieć tylko jednego właściciela.
- 3 Samochód może być przydzielony tylko do jednej naprawy.
- 4 Do naprawy nie może być przydzielony więcej niż jeden samochód.
- 5 Naprawa posiada przydzielonego kierownika, który jest odpowiedzialny za jej przeprowadzenie.
- 6 Każdy pracownik może być kierownikiem wielu napraw.
- 7 Naprawa składa się z jednej lub większej ilości czynności.
- 8 Każda czynność może być wykonywana przez jednego lub wielu pracowników.
- 9 Każdy pracownik może być wykonawcą wielu czynności.
- 10 Każdy pracownik może być członkiem wielu napraw.

3.3 Model logiczny danych



3.3.1 Opis encji i atrybutów oraz ich dziedzin

Klient

- **ID Klienta** – obowiązkowe, cyfrowy identyfikator klienta
- **Imię** – nieobowiązkowe, napis długości nie więcej niż 30 znaków
- **Nazwisko** – obowiązkowe, napis długości nie więcej niż 30 znaków
- **Adres** – obowiązkowe, napis długości nie więcej niż 50 znaków
- **PESEL** – obowiązkowe, napis składający się wyłącznie z cyfr, długości dokładnie 11 znaków

Naprawiany Samochód

- **ID Samochodu** – obowiązkowe, cyfrowy identyfikator samochodu
- **Marka** – obowiązkowe, napis długości nie więcej niż 30 znaków
- **Model** – obowiązkowe, napis długości nie więcej niż 50 znaków
- **Rodzaj Silnika** – obowiązkowe, napis długości nie więcej niż 15 znaków
- **Rejestracja** – obowiązkowe, napis długości nie więcej niż 15 znaków

Naprawa

- **ID Naprawy** – obowiązkowe, cyfrowy identyfikator naprawy
- **Kwota Zapłaty** – obowiązkowe, przedstawienie kwoty
- **Data Oddania** – obowiązkowe, data
- **Data Odbioru** – obowiązkowe, data

Pracownik

- **ID Pracownika** – obowiązkowe, cyfrowy identyfikator pracownika
- **Imię** – nieobowiązkowe, napis długości nie więcej niż 30 znaków
- **Nazwisko** – obowiązkowe, napis długości nie więcej niż 30 znaków
- **Adres** – obowiązkowe, napis długości nie więcej niż 50 znaków
- **PESEL** – obowiązkowe, napis składający się wyłącznie z cyfr, długości dokładnie 11 znaków
- **Data Zatrudnienia** – obowiązkowe, data

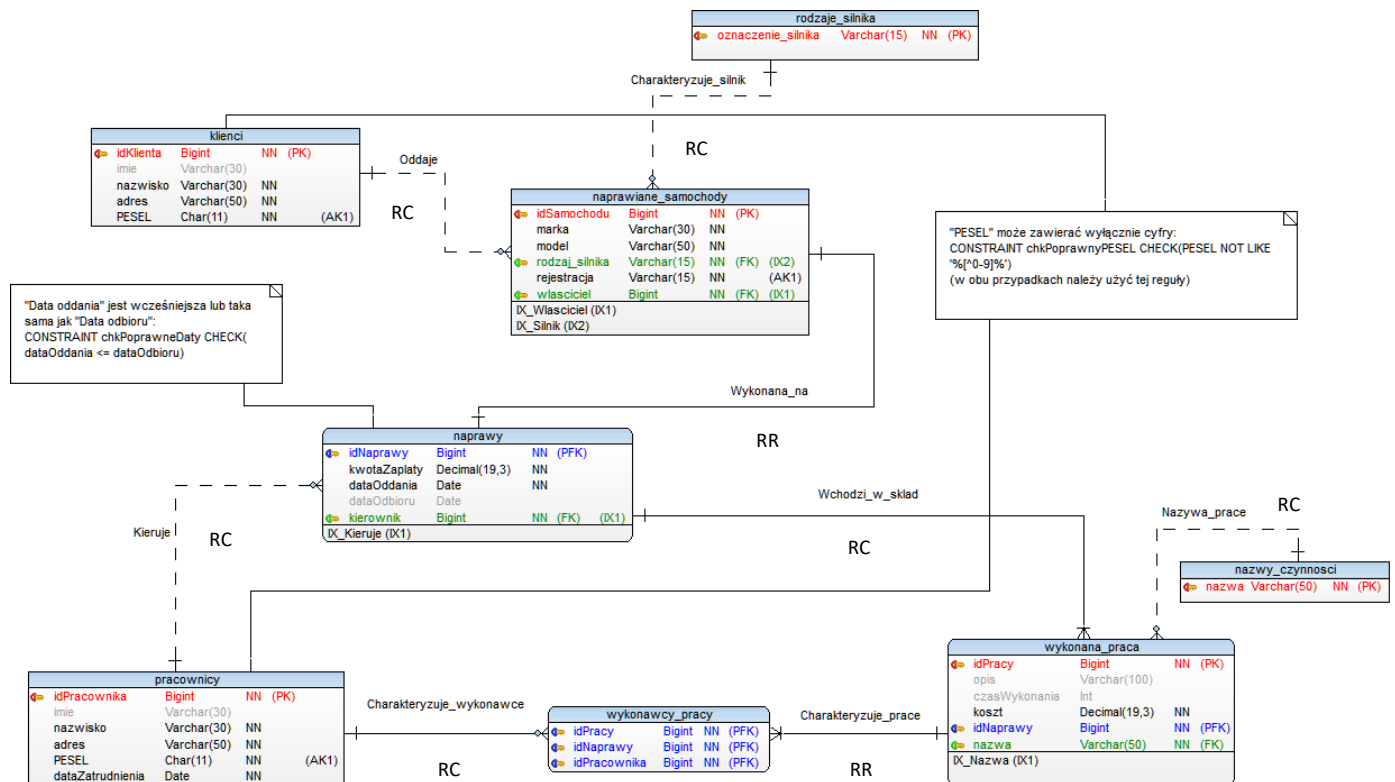
Wykonana Praca

- **ID Pracy** – obowiązkowe, cyfrowy identyfikator wykonanej pracy
- **Nazwa** – obowiązkowe, napis długości nie więcej niż 50 znaków
- **Opis** – nieobowiązkowe, napis długości nie więcej niż 100 znaków
- **Czas wykonania** – nieobowiązkowe, liczba naturalna przedstawiająca ilość godzin (zaokrąglone w górę)
- **Koszt** – obowiązkowe, przedstawienie kwoty

3.3.2 Weryfikacja reguł

- Fakty 1 oraz 2 zamodelowane są przez zależność „Oddaje” między encjami „Klient” i „Naprawiany Samochód”:
 - ✓ Każdy klient może zlecić naprawę wielu samochodów
 - ✓ Każdy samochód może mieć tylko jednego właściciela
- Fakty 3 i 4 zamodelowane są przez zależność „Wykonana na” między encjami „Naprawiany Samochód” i „Naprawa”:
 - ✓ Samochód może być przydzielony tylko do jednej naprawy
 - ✓ Do naprawy nie może być przydzielony więcej niż jeden samochód
- Fakty 5 i 6 zamodelowane są przez zależność „Kieruje” między encjami „Naprawa” i „Pracownik”:
 - ✓ Naprawa posiada przydzielonego kierownika, który jest odpowiedzialny za jej przeprowadzenie
 - ✓ Każdy pracownik może być kierownikiem wielu napraw
- Fakt 7 zamodelowany jest przez zależność „Wchodzi w skład” między encjami „Naprawa” i „Wykonana Praca”:
 - ✓ Naprawa składa się z jednej lub większej ilości czynności
- Fakty 8, 9 i 10 zamodelowane są przez zależność „Wykonuje” między encjami „Pracownik” i „Wykonana Praca”:
 - ✓ Każda czynność może być wykonywana przez jednego lub wielu pracowników
 - ✓ Każdy pracownik może być wykonawcą wielu czynności
 - ✓ Każdy pracownik może być członkiem wielu napraw

3.4 Model fizyczny danych (uniwersalny)



UWAGI:

- Tabela pod nazwą **nazwy_czynnosci** została dodana w celu ułatwienia/przyspieszenia późniejszej pracy, dodanie jej nie jest konieczne. (W przypadku, gdy za każdym razem trzeba byłoby użyć zmiennej wartości, byłoby to zastosowanie błędne). To samo tyczy się tabeli **rodzaje_silnika**. Marka i model nie posiadają dodatkowej tabeli ze względu na zalecenia klienta.
- Wprowadzamy CheckConstraints dla tabeli **naprawy** o treści:
`dataOddania <= dataOdbioru`
Oraz dla tabel **klienci** i **pracownicy** o treści:
`PESEL NOT LIKE '%[^0-9]%'`
- Tabela **naprawy** czerpie swój klucz główny z tabeli **naprawiane_samochody**, i przyjmuje identyfikator **idSamochodu**.
- Kolumny mające przyjmować wartości pieniężne zostały zapisane w typie jako `Decimal(19, 3)`, co sprawdzi się jako odpowiednik liczbowy dla kwoty. W bazach SQL mających dostęp do typu danych np. `money`, jak np. w Microsoft SQL Server, można ten typ zamienić.
- Dodano indeksy dla kolumn **rodzaj_silnika** i **wlasciciel** w tabeli **naprawiane_samochody**, dla kolumny **kierownik** w tabeli **naprawy** oraz kolumny **nazwa** w tabeli **wykonana_praca**.

4 Realizacja Bazy Danych

4.1 Kod SQL tworzący bazę

```
/*
Created: 16/01/2023
Modified: 17/01/2023
Model: Microsoft
Database: MS SQL Server 2019
*/

USE Warsztat
-- Create tables section -----
GO
-- Table klienci

CREATE TABLE [klienci]
(
    [idKlienta] Bigint IDENTITY (1,1) NOT NULL,
    [imie] Varchar(30) NULL,
    [nazwisko] Varchar(30) NOT NULL,
    [adres] Varchar(50) NOT NULL,
    [PESEL] Char(11) NOT NULL,
    CONSTRAINT [CHK_PoprawnyPESELKlient] CHECK (PESEL NOT LIKE '%[^0-9]%')
)
go

-- Add keys for table klienci

ALTER TABLE [klienci] ADD CONSTRAINT [PK_klienci] PRIMARY KEY ([idKlienta])
go

ALTER TABLE [klienci] ADD CONSTRAINT [PESEL_klienci] UNIQUE ([PESEL])
go

-- Table naprawiane_samochody

CREATE TABLE [naprawiane_samochody]
(
    [idSamochodu] Bigint IDENTITY (1,1) NOT NULL,
    [marka] Varchar(30) NOT NULL,
    [model] Varchar(50) NOT NULL,
    [rodzaj_silnika] Varchar(15) NOT NULL,
    [rejestracja] Varchar(15) NOT NULL,
    [wlasciciel] Bigint NOT NULL
)
go

-- Create indexes for table naprawiane_samochody

CREATE INDEX [IX_Wlasciciel] ON [naprawiane_samochody] ([wlasciciel])
go

CREATE INDEX [IX_Silnik] ON [naprawiane_samochody] ([rodzaj_silnika])
go

-- Add keys for table naprawiane_samochody

ALTER TABLE [naprawiane_samochody] ADD CONSTRAINT [PK_naprawiane_samochody] PRIMARY KEY ([idSamochodu])
go

ALTER TABLE [naprawiane_samochody] ADD CONSTRAINT [rejestracja] UNIQUE ([rejestracja])
go

-- Table naprawy

CREATE TABLE [naprawy]
(
    [idNaprawy] Bigint NOT NULL,
    [kwotaZaplaty] Decimal(19,3) NOT NULL,
    [dataOddania] Date DEFAULT GetDate() NOT NULL,
    [dataOdbioru] Date NULL,
    [kierownik] Bigint NOT NULL,
    CONSTRAINT [CHK_PoprawneDaty] CHECK (dataOddania <= dataOdbioru)
)
go
```

```

-- Create indexes for table naprawy

CREATE INDEX [IX_Kieruje] ON [naprawy] ([kierownik])
go

-- Add keys for table naprawy

ALTER TABLE [naprawy] ADD CONSTRAINT [PK_naprawy] PRIMARY KEY ([idNaprawy])
go

-- Table pracownicy

CREATE TABLE [pracownicy]
(
    [idPracownika] Bigint IDENTITY (1,1) NOT NULL,
    [imie] Varchar(30) NULL,
    [nazwisko] Varchar(30) NOT NULL,
    [adres] Varchar(50) NOT NULL,
    [PESEL] Char(11) NOT NULL,
    [dataZatrudnienia] Date NOT NULL,
    CONSTRAINT [CHK_PoprawnyPESELPracownik] CHECK (PESEL NOT LIKE '%[^0-9]%')
)
go

-- Add keys for table pracownicy

ALTER TABLE [pracownicy] ADD CONSTRAINT [PK_pracownicy] PRIMARY KEY ([idPracownika])
go

ALTER TABLE [pracownicy] ADD CONSTRAINT [PESEL_pracownicy] UNIQUE ([PESEL])
go

-- Table wykonana_praca

CREATE TABLE [wykonana_praca]
(
    [idPracy] Bigint IDENTITY (1,1) NOT NULL,
    [opis] Varchar(100) NULL,
    [czasWykonania] Int NULL,
    [koszt] Decimal(19,3) NOT NULL,
    [idNaprawy] Bigint NOT NULL,
    [nazwa] Varchar(50) NOT NULL
)
go

-- Create indexes for table wykonana_praca

CREATE INDEX [IX_Nazwa] ON [wykonana_praca] ([nazwa])
go

-- Add keys for table wykonana_praca

ALTER TABLE [wykonana_praca] ADD CONSTRAINT [PK_wykonana_praca] PRIMARY KEY ([idPracy],[idNaprawy])
go

-- Table wykonawcy_pracy

CREATE TABLE [wykonawcy_pracy]
(
    [idPracy] Bigint NOT NULL,
    [idNaprawy] Bigint NOT NULL,
    [idPracownika] Bigint NOT NULL
)
go

-- Add keys for table wykonawcy_pracy

ALTER TABLE [wykonawcy_pracy] ADD CONSTRAINT [PK_wykonawcy_pracy] PRIMARY KEY ([idPracy],[idNaprawy],[idPracownika])
go

```

```

-- Table nazwy_czynnosci
CREATE TABLE [nazwy_czynnosci]
(
    [nazwa] Varchar(50) NOT NULL
)
go

-- Add keys for table nazwy_czynnosci
ALTER TABLE [nazwy_czynnosci] ADD CONSTRAINT [PK_nazwy_czynnosci] PRIMARY KEY ([nazwa])
go

-- Table rodzaje_silnika
CREATE TABLE [rodzaje_silnika]
(
    [oznaczenie_silnika] Varchar(15) NOT NULL
)
go

-- Add keys for table rodzaje_silnika
ALTER TABLE [rodzaje_silnika] ADD CONSTRAINT [PK_rodzaje_silnika] PRIMARY KEY ([oznaczenie_silnika])
go

-- Create foreign keys (relationships) section -----

ALTER TABLE [naprawiane_samochody] ADD CONSTRAINT [Oddaje] FOREIGN KEY ([wlasosciel]) REFERENCES
[klienci] ([idKlienta]) ON UPDATE CASCADE ON DELETE NO ACTION
go

ALTER TABLE [naprawy] ADD CONSTRAINT [Wykonana_na] FOREIGN KEY ([idNaprawy]) REFERENCES
[naprawiane_samochody] ([idSamochodu]) ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [wykonana_praca] ADD CONSTRAINT [Wchodzi_w_sklad] FOREIGN KEY ([idNaprawy]) REFERENCES
[naprawy] ([idNaprawy]) ON UPDATE CASCADE ON DELETE NO ACTION
go

ALTER TABLE [naprawy] ADD CONSTRAINT [Kieruje] FOREIGN KEY ([kierownik]) REFERENCES
[pracownicy] ([idPracownika]) ON UPDATE CASCADE ON DELETE NO ACTION
go

ALTER TABLE [wykonawcy_pracy] ADD CONSTRAINT [Charakteryzuje_prace] FOREIGN KEY ([idPracy], [idNaprawy]) REFERENCES
[wykonana_praca] ([idPracy], [idNaprawy]) ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [wykonawcy_pracy] ADD CONSTRAINT [Charakteryzuje_wykonawce] FOREIGN KEY ([idPracownika]) REFERENCES
[pracownicy] ([idPracownika]) ON UPDATE CASCADE ON DELETE NO ACTION
go

ALTER TABLE [wykonana_praca] ADD CONSTRAINT [Nazywa_prace] FOREIGN KEY ([nazwa]) REFERENCES
[nazwy_czynnosci] ([nazwa]) ON UPDATE CASCADE ON DELETE NO ACTION
go

ALTER TABLE [naprawiane_samochody] ADD CONSTRAINT [Charakteryzuje_silnik] FOREIGN KEY ([rodzaj_silnika]) REFERENCES
[rodzaje_silnika] ([oznaczenie_silnika]) ON UPDATE CASCADE ON DELETE NO ACTION
go

```


4.1.1 Analiza kodu DDL:SQL

- Ze względu na strukturę bazy, w relacji [Charakteryzuje_prace](#) należy zastosować ograniczenia NO ACTION dla DELETE oraz UPDATE. To samo tyczy się relacji [Wykonana_na](#).
- Dla kolumny [dataOddania](#) została przypisana wartość domyślna, przedstawiająca obecną datę, [dataOdbioru](#) nie posiada takiej wartości, ponieważ w niektórych sytuacjach powinna zostać pusta.

4.1.2 Kod wypełniający bazę danymi testowymi

```
INSERT INTO klienci(imie, nazwisko, adres, PESEL)
VALUES
('Adrian', 'Tomala', 'Kraków, ul.Pomagierów 7', '02300908253'),
('Kamila', 'Starina', 'Kraków, ul.Prądnicka 20', '07896754022'),
('Tomasz', 'Andrzejczyk', 'Wrocław, ul.Matejczyna 32', '08765465489')

INSERT INTO rodzaje_silnika (oznaczenie_silnika)
VALUES
('VQ35DE'),
('CCWA'),
('X20XER')

INSERT INTO naprawiane_samochody (marka, model, rodzaj_silnika, rejestracja, wlasciciel)
VALUES
('Nissan', '350Z', 'VQ35DE', 'KR198XX', 1),
('Audi', 'Q5', 'CCWA', 'KRA4935Y', 2),
('Opel', 'Astra G Sedan', 'X20XER', 'DWR47A5', 3),
('Nissan', '350Z', 'VQ35DE', 'DWR50A2', 1)

INSERT INTO pracownicy (imie, nazwisko, adres, PESEL, dataZatrudnienia)
VALUES
('Norbert', 'Gierczak', 'Kraków, ul.Orkowa 31', '06972695671', '2023-01-17'),
('Karolina', 'Mikołajczyk', 'Kraków, ul.Bajeczna 40', '07657754972', '2022-12-10'),
('Tomasz', 'Zapala', 'Kraków, ul.Malarska 3', '76547887751', '2022-07-16')

GO
SET IDENTITY_INSERT naprawy ON
GO

INSERT INTO naprawy (idNaprawy, kwotaZapłaty, dataOddania, dataOdbioru, kierownik)
VALUES
(1, 15000, '2023-01-06', '2023-01-25', 3),
(2, 500, DEFAULT, '2023-03-20', 3),
(3, 1500, '2022-12-27', '2023-01-27', 2),
(4, 500, DEFAULT, '2023-03-20', 2)

INSERT INTO nazwy_czynnosci (nazwa)
VALUES
('Wymiana klocków hamulcowych'),
('Wymiana skrzyni biegów'),
('Diagnoza'),
('Wymiana zacisków hamulca')

INSERT INTO wykonana_praca (opis, czasWykonania, koszt, idNaprawy, nazwa)
VALUES
('Całkowite wymienienie skrzyni biegów', 17, 15000, 1, 'Wymiana skrzyni biegów'),
(null, 2, 500, 2, 'Diagnoza'),
('Wymiana klocków hamulcowych przy obu przednich kołach', 4, 850, 3, 'Wymiana klocków hamulcowych'),
('Wymiana zacisków hamulca przy obu przednich kołach', 2, 650, 3, 'Wymiana zacisków hamulca'),
(null, 2, 500, 4, 'Diagnoza')

INSERT INTO wykonawcy_pracy (idPracy, idNaprawy, idPracownika)
VALUES
(1, 1, 1),
(1, 1, 2),
(2, 2, 1),
(3, 3, 3),
(4, 3, 3),
(5, 4, 2)
```

4.1.3 Sprawdzenie więzów integralnościowych za pomocą błędnego kodu.

```
--INSERTS THAT WILL THROW ERRORS
/*
-----Naruszenie poprawności dat
INSERT INTO naprawy (idNaprawy, kwotaZapłaty, dataOddania, dataOdbioru, kierownik)
VALUES
(5, 15000, '2023-01-06', '2022-01-25', 3),

-----Naruszenie poprawności PESEL
INSERT INTO klienci(imie, nazwisko, adres, PESEL)
VALUES
( 'Mariusz', 'Tomala', 'Kraków, ul.Pomagierów 7', '0230090825A')

INSERT INTO pracownicy (imie, nazwisko, adres, PESEL, dataZatrudnienia)
VALUES
('Norbert', 'Gierczak', 'Kraków, ul.Orkowa 31', '0697269567B', '2023-01-17')
*/
```

4.1.4 Uwagi do kodu wprowadzającego dane

Zastosowana została komenda `SET IDENTITY_INSERT naprawy ON`. Jest to zabezpieczenie przed ewentualnymi błędami podczas wprowadzania danych, ponieważ tabela naprawy używa klucza obcego jako swój klucz podstawowy.

5 Przykładowe zapytania SQL do bazy danych

5.1 Kwerenda – klienci pochodzący z Krakowa

```
SELECT * FROM klienci
WHERE adres LIKE '%Kraków%'
```

5.2 Kwerenda – ilość samochodów, którą każdy klient oddał do naprawy (zgrupowane po nazwisku i numerze pesel, ułożone alfabetycznie po nazwisku)

```
SELECT COUNT(wlasciciel) AS liczba_samochodow, nazwisko, PESEL
FROM klienci, naprawiane_samochody
WHERE idKlienta = wlasciciel
GROUP BY nazwisko, PESEL
ORDER BY nazwisko ASC
```

5.3 Kwerenda – średnia cena napraw, które zaczęły się w roku 2023

```
SELECT AVG(kwotaZapłaty) AS średnia_cena_naprawy
FROM naprawy
WHERE YEAR(dataOddania) = 2023
```

5.4 Kwerenda – rodzaje silników oraz ich ilość w samochodach, które były naprawiane

```
SELECT rodzaj_silnika, COUNT(rodzaj_silnika) AS ilość_silników
FROM naprawiane_samochody
GROUP BY rodzaj_silnika
```

5.5 Kwerenda – średni czas wykonania czynności, które były wykonane tylko raz (zgrupowane po nazwie i posegregowane rosnąco według czasu).

```
SELECT AVG(czasWykonania) AS sredni_czas, nazwa
FROM wykonana_praca
GROUP BY nazwa
HAVING COUNT(nazwa) < 2
ORDER BY sredni_czas
```

6 Załączniki

- Dokumentacja projektu (plik: dokumentacja.pdf)
- Kod SQL bazy z przykładowymi danymi (plik: baza.sql)