

# Bazy danych i Big Data

## Dokumentacja

Projekt i implementacja relacyjnej bazy danych

Temat projektu:

*Firma transportowa*

Realizator projektu:

Damian Stefanowski

Politechnika Warszawska, Telekomunikacja semestr III

30.04.2021

## Spis treści

<b>1</b>	<b>Zakres i cel projektu</b>	<b>3</b>
1.1	Opis założeń funkcjonalnych projektowanej bazy danych . . . . .	3
<b>2</b>	<b>Definicja systemu</b>	<b>3</b>
2.1	Perspektywy użytkowników . . . . .	3
<b>3</b>	<b>Model konceptualny</b>	<b>5</b>
3.1	Definicja zbiorów encji określonych w projekcie (decyzje projektowe) . . . . .	5
3.2	Ustalenie związków między encjami i ich typów . . . . .	6
3.3	Określenie atrybutów i ich dziedzin . . . . .	8
3.4	Dodatkowe reguły integralnościowe (reguły biznesowe) . . . . .	10
3.5	Klucze kandydujące i główne (decyzje projektowe) . . . . .	11
3.6	Schemat ER na poziomie konceptualnym . . . . .	11
3.7	Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady . . . . .	12
<b>4</b>	<b>Model logiczny</b>	<b>13</b>
4.1	Charakterystyka modelu relacyjnego . . . . .	13
4.2	Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady . . . . .	14
4.3	Proces normalizacji – analiza i przykłady . . . . .	15
4.4	Schemat ER na poziomie modelu logicznego . . . . .	15
4.5	Więzy integralności . . . . .	16
4.6	Proces denormalizacji – analiza i przykłady . . . . .	16
<b>5</b>	<b>Faza fizyczna</b>	<b>17</b>
5.1	Projekt transakcji i weryfikacja ich wykonalności . . . . .	17
5.2	Strojenie bazy danych – dobór indeksów . . . . .	18
5.3	Skrypt SQL zakładający bazę danych . . . . .	18
5.4	Przykłady zapytań i poleceń SQL odnoszących się do bazy danych . . . . .	33

# 1 Zakres i cel projektu

Celem projektu jest zaprojektowanie oraz zaimplementowanie relacyjnej bazy danych. Realizacja obejmuje fazę konceptualną, fazę projektowania logicznego oraz fazę fizyczną. Baza danych zawiera zbiór danych o firmie transportowej.

Oprogramowanie użyte do realizacji projektu:

- Oracle Database 12c
- TOAD Data Modeler 7.3
- SQL Developer
- TOAD for Oracle 14.0

## 1.1 Opis założeń funkcjonalnych projektowanej bazy danych

Realizowany projekt dotyczy firmy transportowej zajmującej się międzynarodowym przewozem towarów po obszarze Europy. W tym celu biuro prowadzi bazę kontrahentów, z którymi na codzień nawiązuje rozmowy oraz negocjacje dążące do pozyskiwania zleceń transportowych. Baza gromadzi wszelkie potrzebne dane na temat kontrahentów oraz pozyskanych zleceń (miejsce, data załadunku/rozładunku), na podstawie których będą wystawiane faktury zawierające dokładne informacje o trasie oraz wycenie usługi. Baza ta jest również podstawą zarządzania taborem, w którego skład wchodzi zestawy (ciągnik oraz naczepa) jak i busy. Dane pojazdu oraz kierowcy, który obsługuje dany pojazd, służą do finalizowania zleceń przez spedytorów, którzy posiadają bezpośredni kontakt z kontrahentami. Wśród osób zatrudnionych specjalne wymagania w kontekście uprawnień i umiejętności stosuje się względem kierowców, którzy muszą posiadać ważne badania lekarskie i prawo jazdy, natomiast w przypadku spedytorów ważna jest umiejętność posługiwania się językiem angielskim. Baza zawiera również dokładniejsze dane na temat pracowników, takie jak: adres, nr. telefonu czy też kwota comiesięcznego wynagrodzenia.

# 2 Definicja systemu

## 2.1 Perspektywy użytkowników

- **Właściciel:**

Posiada dostęp do wszystkich informacji w bazie, może modyfikować każdy element bazy

- **Spedytor:**

Posiada dostęp i prawo modyfikacji danych dotyczących kontrahentów, sfinalizowanych zleceń oraz faktur. Widzi swoje dane osobowe oraz historię wynagrodzeń.

- **Kierowca:**

Widzi swoje dane osobowe, historię wynagrodzeń oraz zadania wynikające z aktualnej pracy, tj. pojazd, który użytkują oraz trasa do której zostali delegowani

**Wyróżniamy następujące funkcjonalności systemu:**

- Podgląd danych o pracownikach oraz ich wynagrodzeniu
- Dodawanie/Usuwanie/Modyfikowanie danych o pracownikach oraz ich wynagrodzeniu
- Podgląd danych o zadaniach wynikających z aktualnej pracy
- Dodawanie/Usuwanie/Modyfikowanie danych o zadaniach wynikających z aktualnej pracy
- Podgląd danych o kontrahentach współpracujących z firmą
- Dodawanie/Usuwanie/Modyfikowanie danych o kontrahentach
- Podgląd danych o przyjętych zleceniach transportowych
- Dodawanie/Usuwanie/Modyfikowanie danych o przyjętych zleceniach transportowych
- Podgląd danych o wystawionych fakturach
- Dodawanie/Usuwanie/Modyfikowanie danych o wystawionych fakturach
- Podgląd danych o posiadanych pojazdach
- Dodawanie/Usuwanie/Modyfikowanie danych o posiadanych pojazdach
- Podgląd danych o biurze
- Dodawanie/Usuwanie/Modyfikowanie danych o biurze
- Podgląd danych o właścicielach
- Dodawanie/Usuwanie/Modyfikowanie danych o właścicielach

## Uprawnienia użytkowników do wyróżnionych funkcjonalności:

Uprawnienia	Właściciel	Spedytor	Kierowca
Podgląd danych o pracownikach oraz ich wynagrodzeniu	+	+	+
Dodawanie/Usuwanie/Modyfikowanie danych o pracownikach oraz ich wynagrodzeniu	+	-	-
Podgląd danych o zadaniach wynikających z aktualnej pracy	+	+	+
Dodawanie/Usuwanie/Modyfikowanie danych o zadaniach wynikających z aktualnej pracy.	+	+	-
Podgląd danych o kontrahentach współpracujących z firmą	+	+	-
Dodawanie/Usuwanie/Modyfikowanie danych o kontrahentach	+	+	-
Podgląd danych o przyjętych zleceniach transportowych.	+	+	-
Dodawanie/Usuwanie/Modyfikowanie danych o przyjętych zleceniach transportowych.	+	+	-
Podgląd danych o wystawionych fakturach	+	+	-
Dodawanie/Usuwanie/Modyfikowanie danych o wystawionych fakturach	+	+	-
Podgląd danych o posiadanych pojazdach	+	+	-
Dodawanie/Usuwanie/Modyfikowanie danych o posiadanych pojazdach	+	-	-
Podgląd danych o biurze	+	+	+
Dodawanie/Usuwanie/Modyfikowanie danych o biurze	+	-	-
Podgląd danych o właścicielach	+	+	+
Dodawanie/Usuwanie/Modyfikowanie danych o właścicielach	+	-	-

## 3 Model konceptualny

### 3.1 Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

- **Biuro** - Główna encja opisująca biuro firmy transportowej
- **Pracownik** - Dokładne dane osobowe pracowników zatrudnionych oraz zwolnionych z firmy
- **Spedytor** - Dodatkowe dane specjalizujące posadę spedytora
- **Kierowca** - Dodatkowe dane specjalizujące posadę kierowcy
- **Kontrahent** - Dane kontrahentów, z którymi współpracuje firma
- **Zlecenie** - Dane zleceń pozyskanych od kontrahentów

- **Pojazd** - Dokładne dane o pojazdach posiadanych przez firmę
- **Bus** - Dodatkowe dane specjalizujące pojazd typu bus
- **Ciągnik** - Dodatkowe dane specjalizujące pojazd typu ciągnik
- **Naczepa** - Dodatkowe dane specjalizujące pojazd typu naczepa

### 3.2 Ustalenie związków między encjami i ich typów

Nazwa relacji	Krotność	Typ uczestnictwa	Opis
Biuro - Pracownik	jeden do wielu	Biuro obowiązkowy Pracownik opcjonalny	Biuro musi istnieć jako encja podstawowa. Pracownik musi podlegać pod biuro, ale biuro nie musi mieć pracowników(np. przy zakładaniu firmy). Biuro może zatrudniać wielu pracowników.
Biuro - Kontrahent	jeden do wielu	Biuro obowiązkowy Kontrahent opcjonalny	Kontrahent musi przynależeć do biura, ale biuro nie musi mieć kontrahenta(np. na początku działalności firmy). Biuro może współpracować z wieloma kontrahentami.
Kontrahent - Zlecenie	jeden do wielu	Kontrahent obowiązkowy Zlecenie opcjonalny	Zlecenie musi być powiązane z kontrahentem, ale kontrahent nie musi być powiązany z żadnym zleceniem(np. gdy trwają rozmowy, ale nie nastąpiło jeszcze oficjalne potwierdzenie). Kontrahent może być powiązany z wieloma zleceniami.
Biuro - Pojazd	jeden do wielu	Biuro obowiązkowy Pojazd opcjonalny	Pojazd musi przynależeć do biura, ale biuro może nie posiadać żadnego pojazdu. Biuro może posiadać wiele pojazdów.
Pojazd - Zlecenie	wiele do wielu	Pojazd opcjonalny Zlecenie opcjonalny	Pojazd może, lecz nie musi, wykonywać wiele zleceń. Zlecenie może, lecz nie musi, być wykonywane przez wiele pojazdów.

Ciągnik - Naczepa	wiele do wielu	Ciągnik opcjonalny Naczepa opcjonalny	Ciągnik może być podpięty pod wiele naczep(nie w jednym momencie), lecz nie musi(np. gdy został świeżo zakupiony). Naczepa może być podpięta pod wiele ciągników(nie w jednym momencie), lecz nie musi(np. gdy została świeżo zakupiona).
Spedytor - Kontrahent	wiele do wielu	Spedytor opcjonalny Kontrahent opcjonalny	Spedytor może kontaktować się z wieloma kontrahentami, lecz nie musi(np. gdy dopiero co został zatrudniony). Kontrahent może, lecz nie musi, kontaktować się z wieloma spedytorami.
Pracownik - Zlecenie	wiele do wielu	Pracownik obowiązkowy Zlecenie opcjonalny	Pracownik może wykonywać wiele zleceń, lecz nie musi wykonywać żadnego(np. gdy jest na urlopie). Zlecenie musi być wykonywane przez pracownika. Zlecenie może być wykonywane przez wielu pracowników.
Pracownik - Pojazd	wiele do wielu	Pracownik opcjonalny Pojazd opcjonalny	Pracownik może użytkować wiele pojazdów(nie w jednym momencie), lecz pojazd nie musi być użytkowany przez pracownika. Pojazd może być użytkowany przez wielu pracowników, lecz pracownik nie musi użytkować pojazdu.

### 3.3 Określenie atrybutów i ich dziedzin

#### Biuro

Nazwa atrybutu	Typ i dziedzina	Obowiązkowość	Opis
Nr_biura	SmallInt	obowiązkowy	Unikatowy identyfikator biura
Nazwa	VarChar(30)	obowiązkowy	Nazwa biura
Adres	VarChar(400)	obowiązkowy	Adres biura
NIP	Character(10)	obowiązkowy	NIP biura
Właściciel	VarChar(300)	obowiązkowy	Właściciel
Data_założenia	Date	obowiązkowy	Data założenia biura

#### Pracownik

Nazwa atrybutu	Typ i dziedzina	Obowiązkowość	Opis
Nr_pracownika	Integer	obowiązkowy	Unikatowy identyfikator pracownika
Imie	VarChar(20)	obowiązkowy	Imię pracownika
Nazwisko	VarChar(30)	obowiązkowy	Nazwisko pracownika
Data_urodzenia	Date	obowiązkowy	Data urodzenia pracownika
Narodowosc	VarChar(30)	obowiązkowy	Narodowość pracownika
PESEL	Character(11)	nieobowiązkowy	PESEL pracownika
Adres	VarChar(400)	obowiązkowy	Adres pracownika
Nr_telefonu	VarChar(14)	obowiązkowy	Numer telefonu pracownika
Plec	PlecD	obowiązkowy	Płeć pracownika
Data_zatrudnienia	Date	obowiązkowy	Data zatrudnienia pracownika
Data_zwolnienia	Date	nieobowiązkowy	Data zwolnienia pracownika
Wynagrodzenie	Money	obowiązkowy	Wynagrodzenie pracownika
Data_waznosci_badania_lek	Date	obowiązkowy	Data ważności badania lekarskiego

#### Kierowca

Nazwa atrybutu	Typ i dziedzina	Obowiązkowość	Opis
Nr_prawa_jazdy	VarChar(10)	obowiązkowy	Numer prawa jazdy
Nr_dowodu	VarChar(10)	obowiązkowy	Nazwa dowodu
Flaga_zagranica	Boolean	obowiązkowy	Zgoda na wyjazd za granicę
Data_waznosci_badania_bhp	Date	obowiązkowy	Data ważności badania BHP
Nr_karty_kierowcy	Character(15)	obowiązkowy	Numer karty kierowcy



### Spedytor

Nazwa atrybutu	Typ i dziedzina	Obowiązkowość	Opis
Flaga_angielski	Boolean	obowiązkowy	Znajomość języka angielskiego
Email	VarChar(50)	obowiązkowy	Adres email
Używana_platforma_gieldowa	VarChar(50)	nieobowiązkowy	Używana platforma giełdowa

### Kontrahent

Nazwa atrybutu	Typ i dziedzina	Obowiązkowość	Opis
Nr_kontrahenta	Integer	obowiązkowy	Unikatowy numer kontrahenta
Nazwa_kontrahenta	VarChar(100)	obowiązkowy	Nazwa kontrahenta
Adres	VarChar(400)	obowiązkowy	Adres kontrahenta
Email	VarChar(50)	obowiązkowy	Adres email

### Zlecenie

Nazwa atrybutu	Typ i dziedzina	Obowiązkowość	Opis
Nr_zlecenia	integer	obowiązkowy	Unikatowy identyfikator zlecenia
Kraj_zaladunku	VarChar(30)	obowiązkowy	Kraj miejsca załadunku
Data_zaladunku	Date	obowiązkowy	Data załadunku
Kraj_rozladunku	VarChar(30)	obowiązkowy	Kraj miejsca rozładunku
Data_rozladunku	Date	obowiązkowy	Data rozładunku
Data_wystawienia	Date	obowiązkowy	Data wystawienia zlecenia
Tonaz_towaru_w_tonach	Decimal(2,2)	obowiązkowy	Tonaż towaru wyrażony w tonach
Faktura	VarChar(400)	obowiązkowy	Faktura

### Pojazd

Nazwa atrybutu	Typ i dziedzina	Obowiązkowość	Opis
Nr_pojazdu	Integer	obowiązkowy	Unikatowy identyfikator pojazdu
Marka	VarChar(30)	obowiązkowy	Marka pojazdu
Model	VarChar(12)	obowiązkowy	Model pojazdu
Rok_produkcji	Date	obowiązkowy	Rok produkcji
VIN	Character(17)	obowiązkowy	Nr. VIN pojazdu
Nr_rejestracyjny	VarChar(7)	obowiązkowy	Numer rejestracyjny
Data_przeglądu_tech	Date	obowiązkowy	Data przeglądu technicznego

### Ciagnik

Nazwa atrybutu	Typ i dziedzina	Obowiązkowość	Opis
Norma_emisji_spalin	NormaEmisjiSpalinD	obowiązkowy	Norma emisji spalin pojazdu
Rodzaj_skrzyni_biegow	RodzajSkrzyniBiegowD	obowiązkowy	Rodzaj skrzyni biegów
Flaga_retarder	Boolean	obowiązkowy	Posiadanie retardera
Pojemnosc_zbiornikow	Integer	obowiązkowy	Nr. Pojemnosc zbiornikow

### Naczepa

Nazwa atrybutu	Typ i dziedzina	Obowiązkowość	Opis
Typ_naczepy	TypNaczepyD	obowiązkowy	Typ naczepy
Flaga_podwieszana_os	Boolean	obowiązkowy	Podwieszana pierwsza oś

### Bus

Nazwa atrybutu	Typ i dziedzina	Obowiązkowość	Opis
Max_waga_tony	Decimal(4,2)	obowiązkowy	Maksymalna ładowność
Dlugosc_paki_metry	Decimal(3,2)	obowiązkowy	Długość paki w metrach
Szerokosc_paki_metry	Decimal(3,2)	obowiązkowy	Szerokość paki w metrach

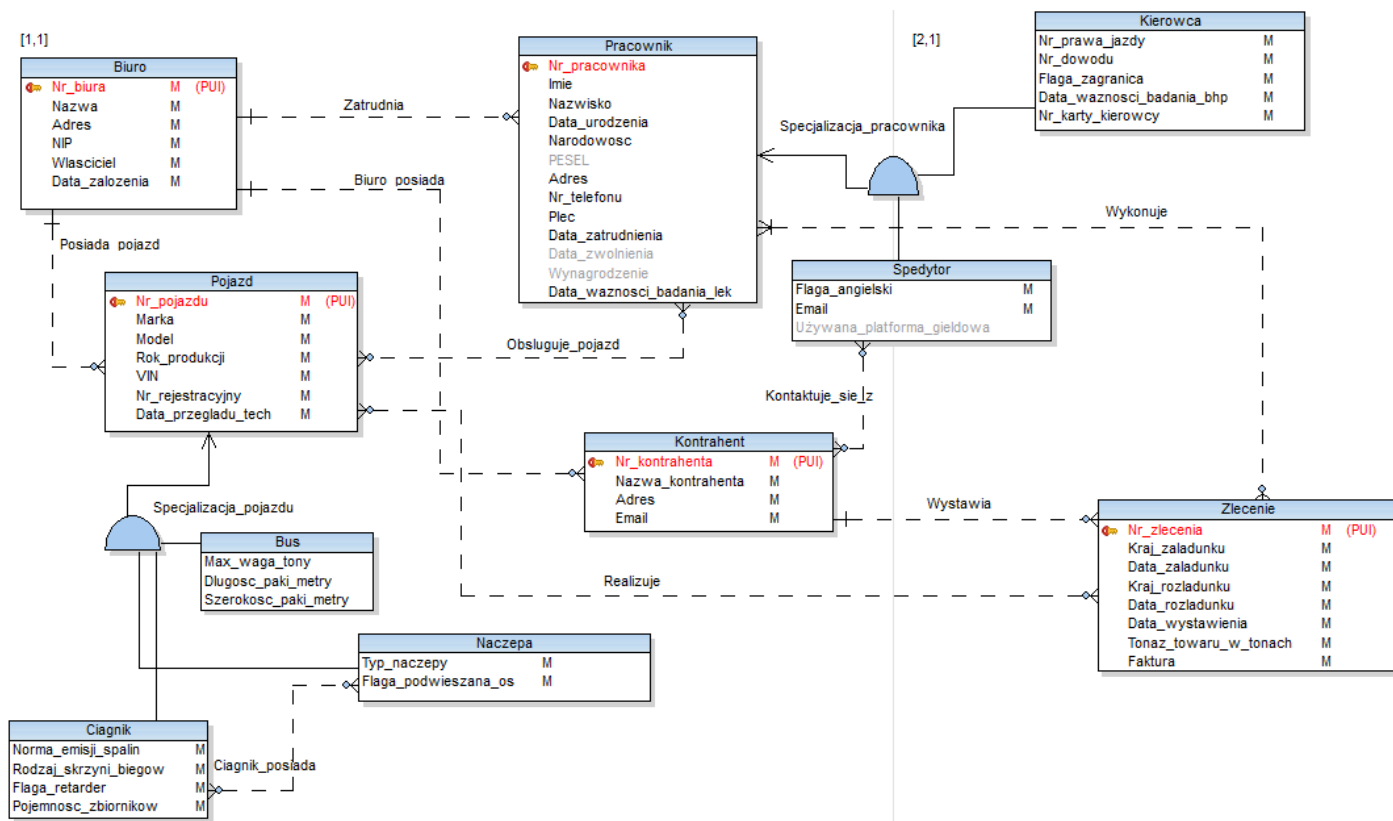
## 3.4 Dodatkowe reguły integralnościowe (reguły biznesowe)

- Pracownik nie może użytkować więcej niż jednego pojazdu w tym samym czasie.
- Ciągnik nie może być podczepiony do więcej niż jednej naczepy w tym samym momencie.
- Data rozładunku na zleceniu nie może być wcześniejsza niż data załadunku.

### 3.5 Klucze kandydujące i główne (decyzje projektowe)

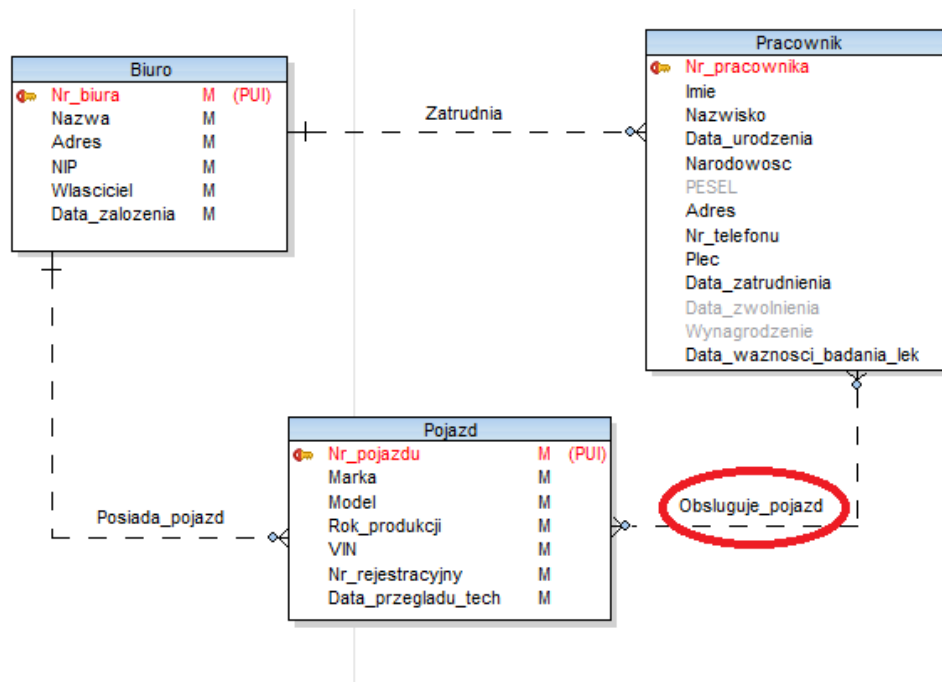
Nazwa encji	Klucz główny	Klucz kandydujący
Biuro	Nr_biura	NIP lub Nazwa
Pracownik	Nr_pracownika	PESEL lub Nr_telefonu
Kierowca	Nr_pracownika	Nr_prawa_jazdy lub Nr_dowodu lub Nr_karty_kierowcy
Spedytor	Nr_pracownika	-
Kontrahent	Nr_kontrahenta	Nazwa_kontrahenta
Zlecenie	Nr_zlecenia	-
Pojazd	Nr_pojazdu	VIN lub Nr_rejestracyjny
Ciagnik	Nr_pojazdu	-
Naczepa	Nr_pojazdu	-
Bus	Nr_pojazdu	-

### 3.6 Schemat ER na poziomie konceptualnym



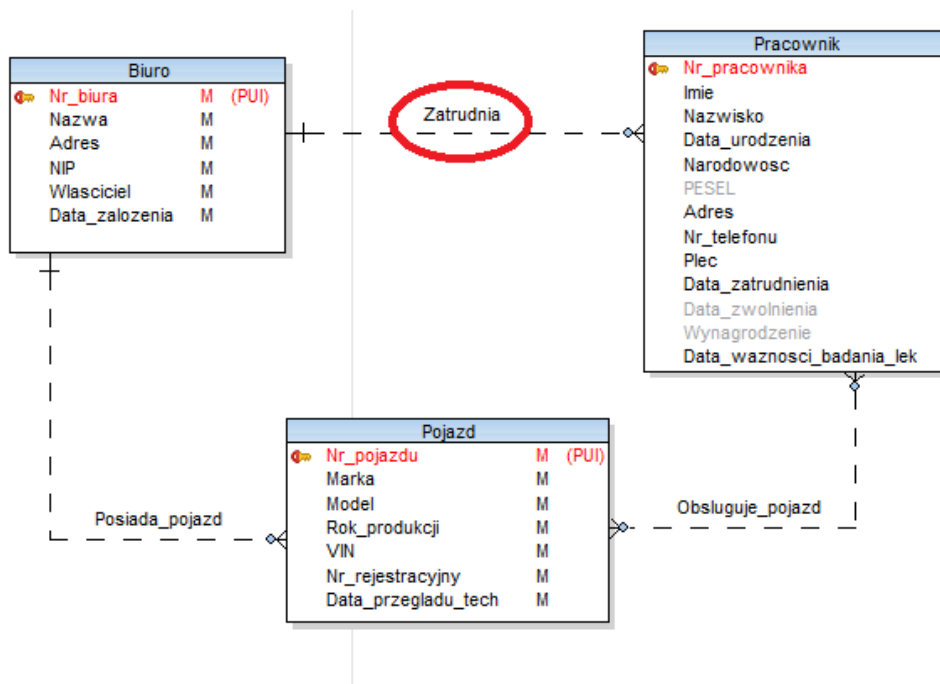
### 3.7 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

Pułapka wachlarzowa:



W projekcie, encja Pracownik oraz encja Pojazd są powiązane z encją Biuro, która zawsze będzie miała krotność jeden. Wtedy powiązanie pomiędzy Pojazdem i Pracownikiem jest niejednoznaczne i mamy do czynienia z pułapką wachlarzową. Rozwiązaniem jest utworzenie związku "Obsluguje\_pojazd" (zaznaczonego na czerwono), który jednoznacznie określi powiązanie pomiędzy pracownikiem, a pojazdem.

### Pułapka szczelinowa:



W projekcie, encja Pracownik jest powiązana z encją Biuro oraz z encją Pojazd. Gdyby nie istniał związek "Zatrudnia" (oznaczony na czerwono), mielibyśmy do czynienia z pułapką szczelinową. Oznaczałoby to, że każdy pracownik obsługuje jakiś pojazd, lecz mamy w firmie pracowników np. spedytor, którzy nie użytkują żadnego pojazdu.

## 4 Model logiczny

### 4.1 Charakterystyka modelu relacyjnego

Konwersja modelu konceptualnego na model relacyjny wprowadziła opisane poniżej zmiany, które zostały wprowadzone automatycznie:

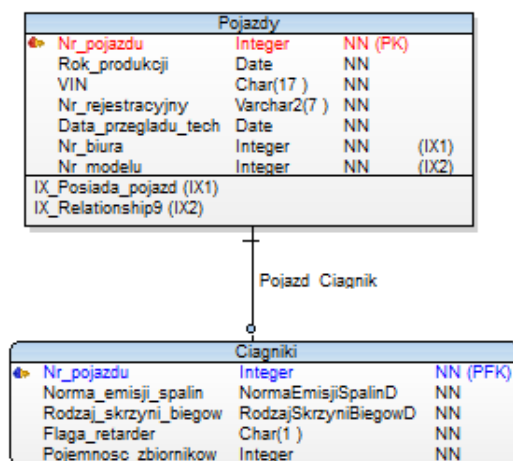
- Związki wiele do wielu zostały zastąpione dwoma związkami jeden do wielu przy użyciu tablicy łączącej, która zawiera klucze główne obu relacji
- Dziedziczenie zostało zastąpione związkiem jeden do jeden. Relacja dziedzicząca otrzymała klucz obcy, który jest kluczem głównym w relacji nadrzędnej
- Relacje otrzymały dodatkowe atrybuty w postaci kluczy obcych
- Typy danych zostały odpowiednio skonwertowane, na typy obsługiwane przez silnik naszej bazy(Oracle Database 12c)
- Folder z regułami(Rules) został zamieniony na folder Check Constraint Rules

Dodatkowe zmiany dokonane po konwersji:

- Zamiana nazwy każdej relacji z liczby pojedynczej na liczbę mnogą
- Odkomentowanie reguł umieszczonych w folderze Check Constraint Rules
- Usunięcie atrybutów, które były polem segmentowym, a następnie utworzenie nowych relacji odzwierciedlających dany atrybut i połączenie relacji za pomocą związku z pierwotną relacją, do której należał
- Przesunięcie klucza obcego na pierwszą pozycję w relacjach, które dziedziczyły po innych relacjach

## 4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

- Modyfikacja krotności związków, które odzwierciedlały dziedziczenie. Przykład:  
Związek "Pojazd.Ciagnik" początkowo miał krotność jeden do wielu, zatem zamieniono go na związek jeden do jednego.

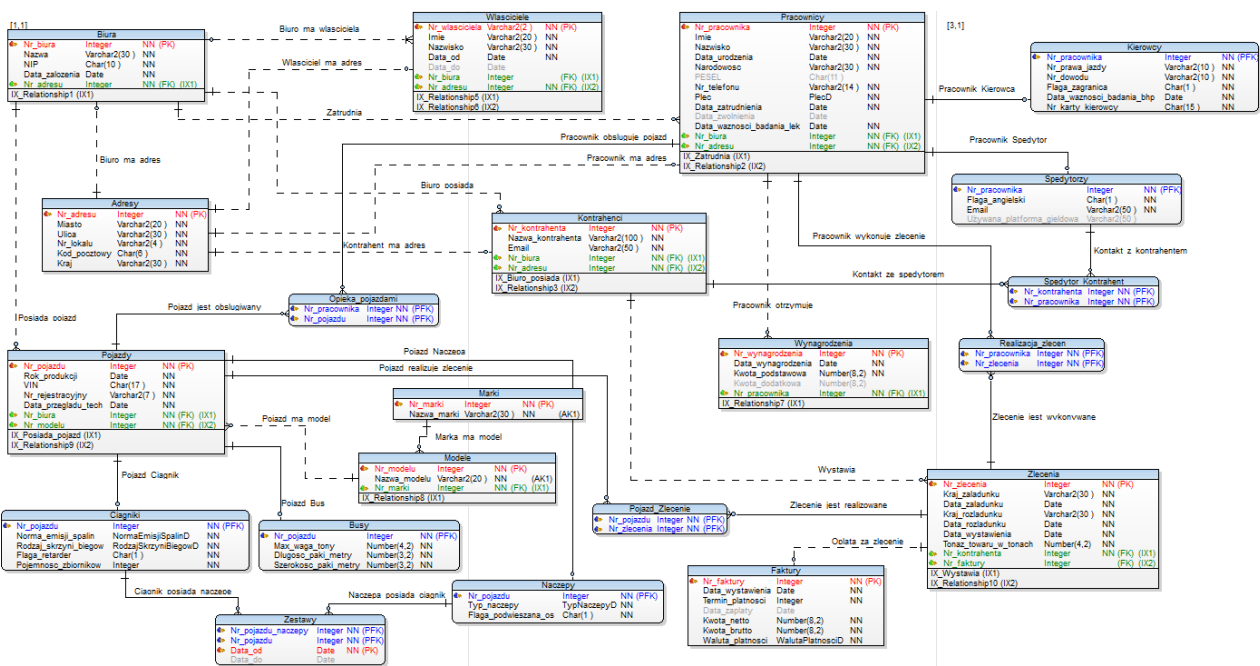


- Usunięcie pól wielowartościowych  
Atrybut Adres stał się osobną relacją. Adres = Miasto, Ulica, Nr. lokalu, Kod pocztowy, Kraj

## 4.3 Proces normalizacji – analiza i przykłady

- Uzyskanie pierwszej postaci normalnej:
  - Uzyskanie wartości atomowej, w każdym atrybucie wszystkich relacji. Przykładem jest atrybut Adres, który jak ukazano powyżej, został przekształcony w osobną relację. Następnie relacja została powiązana z tymi relacjami, które używały atrybutu Adres tj. Biura, Pracownicy, Kontrahenci, Wlasciele.
  - Brak powtarzających się grup. Ten wymóg uzyskano poprzez np. utworzenie relacji Marki oraz Modele.
- Uzyskanie drugiej postaci normalnej:
  - Relacja będąca w pierwszej postaci normalnej, jest równocześnie w drugiej postaci normalnej, jeśli wszystkie jej klucze potencjalne są kluczami prostymi. W naszej bazie, każda relacja zawiera unikatowy identyfikator w postaci "Numeru", który jest kluczem prostym, co implikuje to, że baza jest w drugiej postaci normalnej.
- Uzyskanie trzeciej postaci normalnej:
  - Do uzyskania trzeciej postaci normalnej nie potrzebne były żadne modyfikacje, ponieważ każdy atrybut nie wchodzący w skład żadnego klucza potencjalnego nie jest przechodnio funkcyjnie zależny od żadnego klucza potencjalnego tej relacji. Taką właściwość zagwarantowało nam utworzenie poprawnego modelu konceptualnego.

## 4.4 Schemat ER na poziomie modelu logicznego



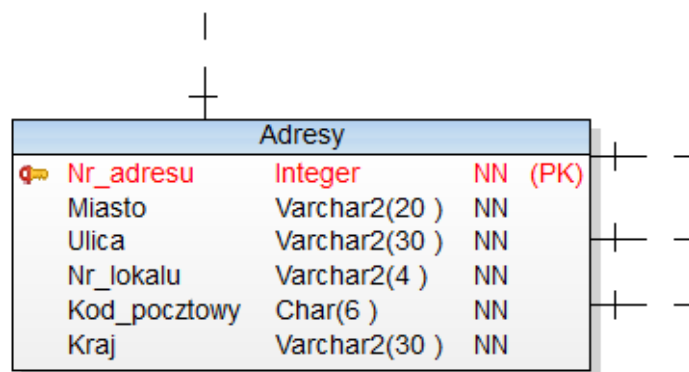
## 4.5 Więzy integralności

- Każdy klucz główny jest unikatowy oraz jest kluczem prostym
- Prawie wszystkie atrybuty nie akceptują wartości NULL, wyjątkiem są takie atrybuty, które w momencie tworzenia rekordu nie są znane np. data zwolnienia pracownika
- Zastosowanie reguł CHECK np. Waluta\_płatności IN ('EUR','PLN','GBP','USD')

## 4.6 Proces denormalizacji – analiza i przykłady

W naszym projekcie nie wykorzystano użycia procesu denormalizacji, jednakże poniżej przedstawiono przykład, w którym mogłaby zostać użyta.

Bardzo prawdopodobne jest, iż często będzie potrzebny adres kontrahenta w celu zaadresowania i wysyłki dokumentacji zleconego transportu. W tym celu można zastosować denormalizację, aby przyspieszyć proces wyszukiwania adresu. Niemniej jednak wcielenie relacji Adresy do relacji Kontrahenci wiąże się z tym, że relacja Adresy będzie musiała również zostać wcielona do relacji Pracownicy, Biura, Właściciele, co spowoduje olbrzymią redundancję danych, przez co denormalizacja może okazać się nieopłacalna.





## 5 Faza fizyczna

### 5.1 Projekt transakcji i weryfikacja ich wykonalności

Uprawnienia	Weryfikacja wykonalności	Wymagane relacje
Podgląd danych o pracownikach oraz ich wynagrodzeniu	Wykonalne	Pracownicy, Adres, Wynagrodzenia
Dodawanie/Usuwanie/Modyfikowanie danych o pracownikach oraz ich wynagrodzeniu	Wykonalne	Pracownicy, Adres, Wynagrodzenia
Podgląd danych o zadaniach wynikających z aktualnej pracy	Wykonalne	Pracownicy, Realizacja_zleceń, Zlecenia
Dodawanie/Usuwanie/Modyfikowanie danych o zadaniach wynikających z aktualnej pracy.	Wykonalne	Pracownicy, Realizacja_zleceń, Zlecenia
Podgląd danych o kontrahentach współpracujących z firmą	Wykonalne	Kontrahenci, Adres
Dodawanie/Usuwanie/Modyfikowanie danych o kontrahentach	Wykonalne	Kontrahenci, Adres
Podgląd danych o przyjętych zleceniach transportowych	Wykonalne	Kontrahenci, Zlecenia
Dodawanie/Usuwanie/Modyfikowanie danych o przyjętych zleceniach transportowych.	Wykonalne	Kontrahenci, Zlecenia
Podgląd danych o wystawionych fakturach	Wykonalne	Faktury, Zlecenia, Kontrahenci
Dodawanie/Usuwanie/Modyfikowanie danych o wystawionych fakturach	Wykonalne	Faktury, Zlecenia, Kontrahenci
Podgląd danych o posiadanych pojazdach	Wykonalne	Pojazdy, Marki, Modele
Dodawanie/Usuwanie/Modyfikowanie danych o posiadanych pojazdach	Wykonalne	Pojazdy, Marki, Modele

Podgląd danych o biurze	Wykonalne	Biura, Adresy
Dodawanie/Usuwanie/Modyfikowanie danych o biurze	Wykonalne	Biura, Adresy
Podgląd danych o właścicielach	Wykonalne	Wlasciciele, Adresy
Dodawanie/Usuwanie/Modyfikowanie danych o właścicielach	Wykonalne	Wlasciciele, Adresy

## 5.2 Strojenie bazy danych – dobór indeksów

Poniżej przedstawiono najważniejsze indeksy utworzone automatycznie:

- Wyszukiwanie pracowników w relacji Wynagrodzenia  
CREATE INDEX IX\_Relationship7 ON Wynagrodzenia (Nr\_pracownika)
- Wyszukiwanie marki w relacji Modele  
CREATE INDEX IX\_Relationship8 ON Modele (Nr\_marki)
- Wyszukiwanie adresów w relacji Pracownicy  
CREATE INDEX IX\_Relationship2 ON Pracownicy (Nr\_adresu)
- Wyszukiwanie modeli w relacji Pojazdy  
CREATE INDEX IX\_Relationship9 ON Pojazdy (Nr\_modelu)
- Wyszukiwanie adresu w relacji Kontrahenci  
CREATE INDEX IX\_Relationship3 ON Kontrahenci (Nr\_adresu)
- Wyszukiwanie Kontrahenta w relacji Zlecenia  
CREATE INDEX IX\_Wystawia ON Zlecenia (Nr\_kontrahenta)
- Wyszukiwanie faktury w relacji Zlecenia  
CREATE INDEX IX\_Relationship10 ON Zlecenia (Nr\_faktury)

## 5.3 Skrypt SQL zakładający bazę danych

```

/*
Created: 26.04.2021
Modified: 29.04.2021
Model: Firma_transportowa
Database: Oracle 12c
*/

```

— Create sequences section —

```
CREATE SEQUENCE BiuraSeq1
```

```
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE AdresySeq1
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE PojazdySeq1
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE ModeleSeq1
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE MarkiSeq1
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE WlascicieleSeq1
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```

CREATE SEQUENCE PracownicySeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

```

```

CREATE SEQUENCE KontrahenciSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

```

```

CREATE SEQUENCE WynagrodzeniaSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

```

```

CREATE SEQUENCE ZleceniaSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

```

```

CREATE SEQUENCE FakturySeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

```

— *Create tables section* —

— *Table Adresy*

```

CREATE TABLE Adresy(
  Nr_adresu Integer NOT NULL,
  Miasto Varchar2(20 ) NOT NULL,

```

```

    Ulica Varchar2(30 ) NOT NULL,
    Nr_lokalu Varchar2(4 ) NOT NULL,
    Kod_pocztowy Char(6 ) NOT NULL,
    Kraj Varchar2(30 ) NOT NULL
)
/

— Add keys for table Adresy

ALTER TABLE Adresy ADD CONSTRAINT PK_Adresy PRIMARY KEY (Nr_adresu)
/

— Table Wlasciciele

CREATE TABLE Wlasciciele(
    Nr_wlasciciela Varchar2(2 ) NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Data_od Date NOT NULL,
    Data_do Date,
    Nr_biura Integer ,
    Nr_adresu Integer NOT NULL
)
/

— Create indexes for table Wlasciciele

CREATE INDEX IX_Relationship5 ON Wlasciciele (Nr_biura)
/

CREATE INDEX IX_Relationship6 ON Wlasciciele (Nr_adresu)
/

— Add keys for table Wlasciciele

ALTER TABLE Wlasciciele ADD CONSTRAINT PK_Wlasciciele PRIMARY KEY (Nr_wlasciciela)
/

— Table Wynagrodzenia

CREATE TABLE Wynagrodzenia(
    Nr_wynagrodzenia Integer NOT NULL,
    Data_wynagrodzenia Date NOT NULL,
    Kwota_podstawowa Number(8,2) NOT NULL,
    Kwota_dodatkowa Number(8,2) ,
    Nr_pracownika Integer NOT NULL
)

```

```

/

— Create indexes for table Wynagrodzenia

CREATE INDEX IX_Relationship7 ON Wynagrodzenia (Nr_pracownika)
/

— Add keys for table Wynagrodzenia

ALTER TABLE Wynagrodzenia ADD CONSTRAINT PK_Wynagrodzenia PRIMARY KEY (Nr_wynagrodzenia)
/

— Table Marki

CREATE TABLE Marki(
    Nr_marki Integer NOT NULL,
    Nazwa_marki Varchar2(30 ) NOT NULL
)
/

— Add keys for table Marki

ALTER TABLE Marki ADD CONSTRAINT PK_Marki PRIMARY KEY (Nr_marki)
/

ALTER TABLE Marki ADD CONSTRAINT Nazwa_marki UNIQUE (Nazwa_marki)
/

— Table Modele

CREATE TABLE Modele(
    Nr_modelu Integer NOT NULL,
    Nazwa_modelu Varchar2(20 ) NOT NULL,
    Nr_marki Integer NOT NULL
)
/

— Create indexes for table Modele

CREATE INDEX IX_Relationship8 ON Modele (Nr_marki)
/

— Add keys for table Modele

ALTER TABLE Modele ADD CONSTRAINT PK_Modele PRIMARY KEY (Nr_modelu)
/

```

```
ALTER TABLE Modele ADD CONSTRAINT Nazwa_modelu UNIQUE (Nazwa_modelu)
/
```

— *Table Biura*

```
CREATE TABLE Biura(
    Nr_biura Integer NOT NULL,
    Nazwa Varchar2(30 ) NOT NULL,
    NIP Char(10 ) NOT NULL,
    Data_zalozenia Date NOT NULL,
    Nr_adresu Integer NOT NULL
)
/
```

— *Create indexes for table Biura*

```
CREATE INDEX IX_Relationship1 ON Biura (Nr_adresu)
/
```

— *Add keys for table Biura*

```
ALTER TABLE Biura ADD CONSTRAINT Biuro_PK PRIMARY KEY (Nr_biura)
/
```

— *Table Pracownicy*

```
CREATE TABLE Pracownicy(
    Nr_pracownika Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Data_urodzenia Date NOT NULL,
    Narodowosc Varchar2(30 ) NOT NULL,
    PESEL Char(11 ),
    Nr_telefonu Varchar2(14 ) NOT NULL,
    Plec Char(256 ) NOT NULL
    CHECK (Plec IN ( 'M' , 'K' )),
    Data_zatrudnienia Date NOT NULL,
    Data_zwolnienia Date,
    Data_waznosci_badania_lek Date NOT NULL,
    Nr_biura Integer NOT NULL,
    Nr_adresu Integer NOT NULL
)
/
```

— *Create indexes for table Pracownicy*

```
CREATE INDEX IX_Zatrudnia ON Pracownicy (Nr_biura)
```

```

/

CREATE INDEX IX_Relationship2 ON Pracownicy (Nr_adresu)
/

— Add keys for table Pracownicy

ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_PK PRIMARY KEY (Nr_pracownika)
/

— Table Kierowcy

CREATE TABLE Kierowcy(
    Nr_pracownika Integer NOT NULL,
    Nr_prawa_jazdy Varchar2(10 ) NOT NULL,
    Nr_dowodu Varchar2(10 ) NOT NULL,
    Flaga_zagranica Char(1 ) NOT NULL,
    Data_waznosci_badania_bhp Date NOT NULL,
    Nr_karty_kierowcy Char(15 ) NOT NULL
)
/

— Add keys for table Kierowcy

ALTER TABLE Kierowcy ADD CONSTRAINT Unique_Identifier1 PRIMARY KEY (Nr_pracownika)
/

— Table Spedytorzy

CREATE TABLE Spedytorzy(
    Nr_pracownika Integer NOT NULL,
    Flaga_angielski Char(1 ) NOT NULL,
    Email Varchar2(50 ) NOT NULL,
    Uywana_platforma_gieldowa Varchar2(50 )
)
/

— Add keys for table Spedytorzy

ALTER TABLE Spedytorzy ADD CONSTRAINT Unique_Identifier2 PRIMARY KEY (Nr_pracownika)
/

— Table Pojazdy

CREATE TABLE Pojazdy(
    Nr_pojazdu Integer NOT NULL,
    Rok_produkcji Date NOT NULL,

```



```

VIN Char(17 ) NOT NULL,
Nr_rejestracyjny Varchar2(7 ) NOT NULL,
Data_przeglądu_tech Date NOT NULL,
Nr_biura Integer NOT NULL,
Nr_modelu Integer NOT NULL
)
/

— Create indexes for table Pojazdy

CREATE INDEX IX_Posiada_pojazd ON Pojazdy (Nr_biura)
/

CREATE INDEX IX_Relationship9 ON Pojazdy (Nr_modelu)
/

— Add keys for table Pojazdy

ALTER TABLE Pojazdy ADD CONSTRAINT Pojazd_PK PRIMARY KEY (Nr_pojazdu)
/

— Table Ciagniki

CREATE TABLE Ciagniki(
Nr_pojazdu Integer NOT NULL,
Norma_emisji_spalin Char(5 ) NOT NULL
CHECK (Norma_emisji_spalin IN ( 'EURO3', 'EURO4', 'EURO5', 'EURO6' )),
Rodzaj_skrzyni_biegow Varchar2(15 ) NOT NULL
CHECK (Rodzaj_skrzyni_biegow IN ( 'AUTOMATYCZNA', 'MANUALNA' )),
Flaga_retarder Char(1 ) NOT NULL,
Pojemnosc_zbiornikow Integer NOT NULL
)
/

— Add keys for table Ciagniki

ALTER TABLE Ciagniki ADD CONSTRAINT Unique_Identifier3 PRIMARY KEY (Nr_pojazdu)
/

— Table Naczepy

CREATE TABLE Naczepy(
Nr_pojazdu Integer NOT NULL,
Typ_naczepy Varchar2(15 ) NOT NULL
CHECK (Typ_naczepy IN ( 'FIRANKA', 'CHŁODNIA', 'LAWETA' )),
Flaga_podwieszana_os Char(1 ) NOT NULL
)

```

```

/

— Add keys for table Naczepy

ALTER TABLE Naczepy ADD CONSTRAINT Naczepa_PK PRIMARY KEY (Nr_pojazdu)
/

— Table Kontrahenci

CREATE TABLE Kontrahenci(
    Nr_kontrahenta Integer NOT NULL,
    Nazwa_kontrahenta Varchar2(100 ) NOT NULL,
    Email Varchar2(50 ) NOT NULL,
    Nr_biura Integer NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

— Create indexes for table Kontrahenci

CREATE INDEX IX_Biuro_posiada ON Kontrahenci (Nr_biura)
/

CREATE INDEX IX_Relationship3 ON Kontrahenci (Nr_adresu)
/

— Add keys for table Kontrahenci

ALTER TABLE Kontrahenci ADD CONSTRAINT Kontrahenci_PK PRIMARY KEY (Nr_kontrahenta)
/

— Table Zlecenia

CREATE TABLE Zlecenia(
    Nr_zlecenia Integer NOT NULL,
    Kraj_zaladunku Varchar2(30 ) NOT NULL,
    Data_zaladunku Date NOT NULL,
    Kraj_rozladunku Varchar2(30 ) NOT NULL,
    Data_rozladunku Date NOT NULL,
    Data_wystawienia Date NOT NULL,
    Tonaz_towaru_w_tonach Number(4,2) NOT NULL,
    Nr_kontrahenta Integer NOT NULL,
    Nr_faktury Integer
)
/

— Create indexes for table Zlecenia

```

```

CREATE INDEX IX_Wystawia ON Zlecenia (Nr_kontrahenta)
/

CREATE INDEX IX_Relationship10 ON Zlecenia (Nr_faktury)
/

— Add keys for table Zlecenia

ALTER TABLE Zlecenia ADD CONSTRAINT Zlecenie_PK PRIMARY KEY (Nr_zlecenia)
/

— Table Busy

CREATE TABLE Busy(
    Nr_pojazdu Integer NOT NULL,
    Max_waga_tony Number(4,2) NOT NULL,
    Dlugosc_paki_metry Number(3,2) NOT NULL,
    Szerokosc_paki_metry Number(3,2) NOT NULL
)
/

— Add keys for table Busy

ALTER TABLE Busy ADD CONSTRAINT Unique_Identifier5 PRIMARY KEY (Nr_pojazdu)
/

— Table Opieka_pojazdami

CREATE TABLE Opieka_pojazdami(
    Nr_pracownika Integer NOT NULL,
    Nr_pojazdu Integer NOT NULL
)
/

— Table Zestawy

CREATE TABLE Zestawy(
    Nr_pojazdu_naczepy Integer NOT NULL,
    Nr_pojazdu Integer NOT NULL,
    Data_od Date NOT NULL,
    Data_do Date
)
/

— Table Spedytor_Kontrahent

```

```

CREATE TABLE Spedytor_Kontrahent(
    Nr_kontrahenta Integer NOT NULL,
    Nr_pracownika Integer NOT NULL
)
/

```

— *Table Pojazd\_Zlecenie*

```

CREATE TABLE Pojazd_Zlecenie(
    Nr_pojazdu Integer NOT NULL,
    Nr_zlecenia Integer NOT NULL
)
/

```

— *Table Realizacja\_zleceń*

```

CREATE TABLE Realizacja_zleceń(
    Nr_pracownika Integer NOT NULL,
    Nr_zlecenia Integer NOT NULL
)
/

```

— *Table Faktury*

```

CREATE TABLE Faktury(
    Nr_faktury Integer NOT NULL,
    Data_wystawienia Date NOT NULL,
    Termin_platnosci Integer NOT NULL,
    Data_zaplaty Date,
    Kwota_netto Number(8,2) NOT NULL,
    Kwota_brutto Number(8,2) NOT NULL,
    Waluta_platnosci Char(3 ) NOT NULL
    CHECK (Waluta_platnosci IN ( 'EUR', 'PLN', 'GBP', 'USD'))
)
/

```

— *Add keys for table Faktury*

```

ALTER TABLE Faktury ADD CONSTRAINT PK_Faktury PRIMARY KEY (Nr_faktury)
/

```

— *Trigger for sequence BiuraSeq1 for column Nr\_biura in table Biura* —————

```

CREATE OR REPLACE TRIGGER ts_Biura_BiuraSeq1 BEFORE INSERT
ON Biura FOR EACH ROW
BEGIN
    :new.Nr_biura := BiuraSeq1.nextval;
END;

```

```

/
CREATE OR REPLACE TRIGGER tsu_Biura_BiuraSeq1 AFTER UPDATE OF Nr_biura
ON Biura FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot_update_column_Nr_biura_in_table_Biura_as_it
    _uses_sequence. ');
END;
/

-- Trigger for sequence PracownicySeq1 for column Nr_pracownika in table Pracownicy -----
CREATE OR REPLACE TRIGGER ts_Pracownicy_PracownicySeq1 BEFORE INSERT
ON Pracownicy FOR EACH ROW
BEGIN
    :new.Nr_pracownika := PracownicySeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Pracownicy_PracownicySeq1 AFTER UPDATE OF Nr_pracownika
ON Pracownicy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot_update_column_Nr_pracownika_in_table_Pracownicy
    _as_it_uses_sequence. ');
END;
/

-- Trigger for sequence PojazdySeq1 for column Nr_pojazdu in table Pojazdy -----
CREATE OR REPLACE TRIGGER ts_Pojazdy_PojazdySeq1 BEFORE INSERT
ON Pojazdy FOR EACH ROW
BEGIN
    :new.Nr_pojazdu := PojazdySeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Pojazdy_PojazdySeq1 AFTER UPDATE OF Nr_pojazdu
ON Pojazdy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot_update_column_Nr_pojazdu_in_table_Pojazdy_as_it
    _uses_sequence. ');
END;
/

-- Trigger for sequence KontrahenciSeq1 for column Nr_kontrahenta in table Kontrahenci -----
CREATE OR REPLACE TRIGGER ts_Kontrahenci_KontrahenciSeq1 BEFORE INSERT
ON Kontrahenci FOR EACH ROW
BEGIN
    :new.Nr_kontrahenta := KontrahenciSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Kontrahenci_KontrahenciS.0 AFTER UPDATE OF Nr_kontrahenta

```

```

ON Kontrahenci FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot_update_column_Nr_kontrahenta_in_table_Kontrahenci_as_it
uses_sequence. ');
END;
/

-- Trigger for sequence ZleceniaSeq1 for column Nr_zlecenia in table Zlecenia -----
CREATE OR REPLACE TRIGGER ts_Zlecenia_ZleceniaSeq1 BEFORE INSERT
ON Zlecenia FOR EACH ROW
BEGIN
    :new.Nr_zlecenia := ZleceniaSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Zlecenia_ZleceniaSeq1 AFTER UPDATE OF Nr_zlecenia
ON Zlecenia FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot_update_column_Nr_zlecenia_in_table_Zlecenia_as_it
uses_sequence. ');
END;
/

-- Trigger for sequence AdresySeq1 for column Nr_adresu in table Adresy -----
CREATE OR REPLACE TRIGGER ts_Adresy_AdresySeq1 BEFORE INSERT
ON Adresy FOR EACH ROW
BEGIN
    :new.Nr_adresu := AdresySeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Adresy_AdresySeq1 AFTER UPDATE OF Nr_adresu
ON Adresy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot_update_column_Nr_adresu_in_table_Adresy_as_it
uses_sequence. ');
END;
/

-- Trigger for sequence WlasciecieSeq1 for column Nr_wlasciciela in table Wlasciecie -----
CREATE OR REPLACE TRIGGER ts_Wlasciecie_WlasciecieSeq1 BEFORE INSERT
ON Wlasciecie FOR EACH ROW
BEGIN
    :new.Nr_wlasciciela := WlasciecieSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wlasciecie_WlasciecieS_0 AFTER UPDATE OF Nr_wlasciciela
ON Wlasciecie FOR EACH ROW
BEGIN

```

```

    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_wlasciciela in table Wlasciciele as
    it uses sequence. ');
END;
/

-- Trigger for sequence WynagrodzeniaSeq1 for column Nr_wynagrodzenia in table Wynagrodzenia -----
CREATE OR REPLACE TRIGGER ts_Wynagrodzenia_Wynagrodzen_0 BEFORE INSERT
ON Wynagrodzenia FOR EACH ROW
BEGIN
    :new.Nr_wynagrodzenia := WynagrodzeniaSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wynagrodzenia_Wynagrodze_0 AFTER UPDATE OF Nr_wynagrodzenia
ON Wynagrodzenia FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_wynagrodzenia in table Wynagrodzenia
    as it uses sequence. ');
END;
/

-- Trigger for sequence MarkiSeq1 for column Nr_marki in table Marki -----
CREATE OR REPLACE TRIGGER ts_Marki_MarkiSeq1 BEFORE INSERT
ON Marki FOR EACH ROW
BEGIN
    :new.Nr_marki := MarkiSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Marki_MarkiSeq1 AFTER UPDATE OF Nr_marki
ON Marki FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_marki in table Marki as it uses
    sequence. ');
END;
/

-- Trigger for sequence ModeleSeq1 for column Nr_modelu in table Modele -----
CREATE OR REPLACE TRIGGER ts_Modele_ModeleSeq1 BEFORE INSERT
ON Modele FOR EACH ROW
BEGIN
    :new.Nr_modelu := ModeleSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Modele_ModeleSeq1 AFTER UPDATE OF Nr_modelu
ON Modele FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_modelu in table Modele as it uses
    sequence. ');

```

```

END;
/

-- Trigger for sequence FakturySeq1 for column Nr_faktury in table Faktury -----
CREATE OR REPLACE TRIGGER ts_Faktury_FakturySeq1 BEFORE INSERT
ON Faktury FOR EACH ROW
BEGIN
    :new.Nr_faktury := FakturySeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Faktury_FakturySeq1 AFTER UPDATE OF Nr_faktury
ON Faktury FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_faktury in table Faktury as it uses
sequence. ');
END;
/

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

ALTER TABLE Pracownicy ADD FOREIGN KEY (Nr_biura) REFERENCES Biura (Nr_biura)
/

ALTER TABLE Pojazdy ADD FOREIGN KEY (Nr_biura) REFERENCES Biura (Nr_biura)
/

ALTER TABLE Kontrahenci ADD FOREIGN KEY (Nr_biura) REFERENCES Biura (Nr_biura)
/

ALTER TABLE Zlecenia ADD FOREIGN KEY (Nr_kontrahenta) REFERENCES Kontrahenci (Nr_kontrahenta)
/

ALTER TABLE Biura ADD FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)
/

ALTER TABLE Pracownicy ADD FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)

```



/

```
ALTER TABLE Kontrahenci ADD FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)
```

/

```
ALTER TABLE Wlasciciele ADD FOREIGN KEY (Nr_biura) REFERENCES Biura (Nr_biura)
```

/

```
ALTER TABLE Wlasciciele ADD FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)
```

/

```
ALTER TABLE Wynagrodzenia ADD FOREIGN KEY (Nr_pracownika) REFERENCES Pracownicy (Nr_pracownika)
```

/

```
ALTER TABLE Modele ADD FOREIGN KEY (Nr_marki) REFERENCES Marki (Nr_marki)
```

/

```
ALTER TABLE Pojazdy ADD FOREIGN KEY (Nr_modelu) REFERENCES Modele (Nr_modelu)
```

/

```
ALTER TABLE Zlecenia ADD FOREIGN KEY (Nr_faktury) REFERENCES Faktury (Nr_faktury)
```

/

## 5.4 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

- Podgląd danych pracownika pracującego jako spedytor oraz jego wynagrodzenia

```
SELECT pr.imie, pr.nazwisko, wy.Kwota_podstawowa, wy.Kwota_dodatkowa, wy.Data_wynagrodzenia
FROM Pracownicy pr
INNER JOIN Wynagrodzenia wy
ON pr.Nr_pracownika = wy.Nr_pracownika
INNER JOIN Spedytorzy sp
ON pr.Nr_pracownika = sp.Nr_pracownika
```

IMIE	NAZWISKO	KWOTA_PODSTAWOWA	KWOTA_DODATKOWA	DATA_WYNAGRODZENIA
Marcin	Zielinski	5000	0	01/04/2021

- Podgląd danych pracownika, aktualnej trasy(wykonywane zadanie) oraz nr. rejestracyjnego pojazdu, który użytkuje

```

SELECT pr.imie, pr.nazwisko, zl.Kraj_zaladunku, zl.Data_zaladunku,
zl.Kraj_rozladunku, zl.Data_rozladunku, po.Nr_rejestracyjny
FROM Pracownicy pr
INNER JOIN Realizacja_zleceń rz
ON pr.Nr_pracownika = rz.Nr_pracownika
INNER JOIN Zlecenia zl
ON rz.Nr_zlecenia = zl.Nr_zlecenia
INNER JOIN Opieka_pojazdami op
ON pr.Nr_pracownika = op.Nr_pracownika
INNER JOIN Pojazdy po
ON op.Nr_pojazdu = po.Nr_pojazdu

```

IMIE	NAZWISKO	KRAJ_ZALADUNKU	DATA_ZALADUNKU	KRAJ_ROZLADUNKU	DATA_ROZLADUNKU	NR_REJESTRACYJNY
Adam	Zubrzycki	Niemcy	30/04/2021	Polska	03/05/2021	WA89973

- Podgląd liczby zleceń zawartych z kontrahentami

```

SELECT ko.Nazwa_kontrahenta, COUNT(*) AS ilosc_zleceń
FROM Kontrahenci ko, Zlecenia zl
WHERE ko.Nr_kontrahenta = zl.Nr_kontrahenta
GROUP BY ko.Nazwa_kontrahenta
ORDER BY ilosc_zleceń DESC

```

NAZWA_KONTRAHENTA	ILOSC_ZLECEN
Amazon DE	4
BETONEX	1

- Liczba zleceń, do których nie została jeszcze wystawiona faktura

```

SELECT COUNT(*) AS ilosc_zleceń FROM Zlecenia zl
WHERE zl.Nr_faktury IS NULL

```

ILOSC_ZLECEN
4

- Podgląd danych adresowych kontrahenta o nazwie BETONEX

```
SELECT ko.Nazwa_kontrahenta, ad.ULICA, ad.NR_LOKALU,
ad.KOD_POCZTOWY, ad.MIASTO, ad.KRAJ
FROM Kontrahenci ko
INNER JOIN Adresy ad
ON ko.Nr_adresu = ad.Nr_adresu
WHERE ko.Nazwa_kontrahenta = 'BETONEX'
```

NAZWA_KONTRAHENTA	ULICA	NR_LOKALU	KOD_POCZTOWY	MIASTO	KRAJ
BETONEX	Wawelska	109	79-987	Krakow	Polska

- Podgląd danych pojazdu oraz liczby dni do końca ważności badania technicznego

```
SELECT po.NR_REJESTRACYJNY, ma.NAZWA_MARKI, mo.NAZWA_MODELU,
trunc(po.DATA_PRZEGLEDU_TECH - SYSDATE) AS liczba_dni
FROM Pojazdy po, Modele mo, Marki ma
WHERE po.NR_MODELU = mo.NR_MODELU
```

NR_REJESTRACYJNY	NAZWA_MARKI	NAZWA_MODELU	LICZBA_DNI
WA89973	DAF	XF 105	244
WA87677	DAF	XF 105	152