

Jan Grzybowski, Piotr Jabłoński

# Projekt BDBT

## Tramwajowe przedsiębiorstwo miejskie

19 listopada 2020

### Spis treści

<b>1. Zakres i cel projektu</b>	2
1.1. Założenia projektu	2
<b>2. Definicja systemu</b>	2
2.1. Perspektywy użytkowników	2
2.1.1. Administrator	2
2.1.2. Księgowa	2
2.1.3. Pasażer	2
2.1.4. Pracownik	2
2.1.5. Kierownik	2
<b>3. Model konceptualny</b>	3
3.1. Definicja zbiorów encji określonych w projekcie (decyzje projektowe)	3
3.2. Ustalenie związków między encjami i ich typów	5
3.3. Klucze kandydujące i główne (decyzje projektowe)	6
3.4. Schemat ER na poziomie konceptualnym	7
3.5. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady	8
3.5.1. Przykład pułapki szczelinowej	8
3.5.2. Przykład pułapki wachlarzowej	8
<b>4. Model logiczny</b>	8
4.1. Charakterystyka modelu relacyjnego	8
4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym – przykłady	8
4.2.1. Usunięcie związków wiele do wielu	8
4.2.2. Usunięcie specjalizacji	9
4.3. Proces normalizacji – analiza i przykłady	9
4.3.1. 1 postać normalna	9
4.3.2. 2 postać normalna	9
4.3.3. 3 postać normalna	10
4.4. Schemat ER na poziomie modelu logicznego	10
4.5. Więzy integralności	12
<b>5. Faza fizyczna</b>	12
5.1. Projekt transakcji i weryfikacja ich wykonywalności	12
5.2. Strojenie bazy danych – dobór indeksów	12
5.3. Skrypt SQL zakładający bazę danych	13
5.4. Skrypt SQL uzupełniający bazę o przykładowe dane	25
5.5. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych	26
<b>6. Bibliografia</b>	27

## 1. Zakres i cel projektu

Celem projektu jest zaprojektowanie relacyjnej bazy danych na poziomie konceptualnym i logicznym oraz jej fizyczna implementacja. Baza danych obsługuje **tramwajowe przedsiębiorstwo miejskie** i będzie oparta na rozwiązaniach firmy Oracle, a wykorzystanym językiem będzie SQL.

### 1.1. Założenia projektu

Realizowany projekt dotyczy infrastruktury tramwajowego przedsiębiorstwa miejskiego (Zarząd Transportu Miejskiego). ZTM może posiadać zajezdnie oraz przystanki i połączenia pomiędzy nimi. W bazie przechowywane są informacje na temat posiadanych tramwajów i rozkładu jazdy. Poza tym w bazie zawarte są również dane pracowników, m.in. ich stanowisko oraz wynagrodzenie.

## 2. Definicja systemu

### Funkcjonalność systemu zakłada:

1. Dodawanie, usuwanie i modyfikacja biur ZTM
2. Podgląd danych ZTM
3. Dodawanie, usuwanie, modyfikacja pracownika i jego danych
4. Podgląd danych pracownika
5. Dodawanie, usuwanie, modyfikacja tramwaju i jego danych
6. Podgląd danych tramwaju
7. Dodawanie, usuwanie, modyfikacja przystanku i jego danych
8. Podgląd danych przystanku
9. Dodawanie, usuwanie, modyfikacja zajezdni i jej danych
10. Podgląd danych zajezdni
11. Dodawanie, usuwanie, modyfikacja linii i jej danych
12. Podgląd danych linii
13. Dodawanie, usuwanie, modyfikacja stanowiska i jego danych
14. Podgląd danych stanowiska
15. Dodawanie, usuwanie, modyfikacja marki tramwaju i jej danych
16. Podgląd danych marki tramwaju
17. Dodawanie, usuwanie, modyfikacja modelu tramwaju i jego danych
18. Podgląd danych modelu tramwaju
19. Dodawanie, usuwanie, modyfikacja rozkładu jazdy i jego danych
20. Podgląd danych rozkładu jazdy
21. Podgląd listy wynagrodzeń

### 2.1. Perspektywy użytkowników

#### 2.1.1. Administrator

Posiada dostęp do wszystkich funkcjonalności systemu i modyfikacji bazy danych. Ma uprawnienia administratora bazy danych Oracle.

#### 2.1.2. Księgowa

Odpowiada za finanse- ma dostęp do danych personalnych pracowników, informacji o ich wynagrodzeniu.

#### 2.1.3. Pasażer

Ma dostęp do adresów przystanków, zajezdni, ZTM oraz do rozkładu jazdy.

#### 2.1.4. Pracownik

Posiada dostęp do swoich danych oraz danych tramwaju, jeżeli jest jego opiekunem.

#### 2.1.5. Kierownik

Posiada dostęp do wszystkich danych związanych z placówką, może je modyfikować.

### 3. Model konceptualny

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

Poniżej przedstawiamy w formie tabel charakterystykę encji, jej atrybutów tj. Nazwy(Name); Typu danych(Data Type); Parametr typu danych(Data Type Param 1); Dziedziny(Domain); Informacji, czy jest kluczem głównym(Primary Identifier); Informacji, czy atrybut jest obowiązkowy(Mandatory) oraz Opisu(Description).

**Encja ZTM** - istnieje tylko jeden obiekt tej encji, definiuje biuro ZTM.

Name	Domain	Data Type	Data Type Param 1	Primary Identifier	Mandatory	Description
Nr_biura		SmallInt		PRAWDA	PRAWDA	Numer biura
Adres		VarChar(%p1)	200	FAŁSZ	PRAWDA	Adres ZTM, pole segmentowe<miasto,ulica, numer lokalu, kod pocztowy, poczta>
Nazwa		VarChar(%p1)	50	FAŁSZ	PRAWDA	Nazwa biura ZTM
Miasto		VarChar(%p1)	50	FAŁSZ	PRAWDA	Miasto w którym znajduje się nasze przedsiębiorstwo
Data_zalozenia		Date		FAŁSZ	PRAWDA	Data założenia biura ZTM

**Encja Pracownik** - istnieje wiele obiektów tej encji, definiuje pracownika. Została zdefiniowana dziedzina PlecD, typu char o długości 1. Możliwe znaki do wyboru to K(Kobieta) lub M(Mężczyzna).

Name	Domain	Data Type	Data Type Param 1	Primary Identifier	Mandatory	Description
Nr_pracownika		Integer		PRAWDA	PRAWDA	Unikatowy numer pracownika
Imię		VarChar(%p1)	20	FAŁSZ	PRAWDA	Imię pracownika
Nazwisko		VarChar(%p1)	40	FAŁSZ	PRAWDA	Nazwisko pracownika
Data_urodzenia		Date		FAŁSZ	PRAWDA	Data urodzenia pracownika
Plec	PlecD		1	FAŁSZ	FAŁSZ	Płeć pracownika
Pesel		Character(%p1)	11	FAŁSZ	FAŁSZ	Numer pesel pracownika
Adres		VarChar(%p1)	400	FAŁSZ	PRAWDA	Adres pracownika, pole segmentowe{miasto, ulica, numer lokalu, kod pocztowy, poczta}
Data_zatrudnienia		Date		FAŁSZ	PRAWDA	Data zatrudnienia pracownika
Stanowisko		VarChar(%p1)	30	FAŁSZ	PRAWDA	Stanowisko pracownika
Data_zwolnienia		Date		FAŁSZ	FAŁSZ	Data zwolnienia pracownika
Wynagrodzenie		Money		FAŁSZ	FAŁSZ	Wynagrodzenie pracownika, pole wielowartościowe, pole segmentowe<data wynagrodzenia, kwota wynagrodzenia>

**Encja Motorniczy** - specjalizacja encji pracownik, może istnieć wiele obiektów tej encji, definiuje motorniczego.

Name	Domain	Data Type	Data Type Param 1	Primary Identifier	Mandatory	Description
Data_badania		Date		FAŁSZ	PRAWDA	Data ważności badania lekarskiego
Nr_licencji_tramwajowej		Integer		FAŁSZ	PRAWDA	Numer licencji pozwalającej na prowadzenie tramwajów w firmie
Data_waznosci_licencji		Date		FAŁSZ	PRAWDA	Data ważności licencji tramwajowej

**Encja Tramwaj** - istnieje wiele obiektów tej encji, definiuje tramwaj.

Name	Domain	Data Type	Data Type Param 1	Primary Identifier	Mandatory	Description
Nr_pojazdu		Integer		PRAWDA	PRAWDA	Unikatowy numer pojazdu
Marka		VarChar(%p1)	50	FAŁSZ	PRAWDA	Marka pojazdu
Model		VarChar(%p1)	50	FAŁSZ	PRAWDA	Model pojazdu
Rok_produkcji		SmallInt		FAŁSZ	PRAWDA	Rok produkcji pojazdu
Liczba_drzwi		SmallInt		FAŁSZ	PRAWDA	Liczba drzwi pojazdu
Liczba_miejsc_siedzacych		SmallInt		FAŁSZ	PRAWDA	Liczba miejsc siedzących w pojeździe
Data_przeglądu		Date		FAŁSZ	PRAWDA	Data kolejnego przeglądu
Liczba_miejsc_stojacych		SmallInt		FAŁSZ	PRAWDA	Liczba miejsc stojących w tramwaju
Czy_biletomat		Boolean		FAŁSZ	PRAWDA	Czy tramwaj posiada biletomat
Czy_niskopodlogowy		Bigint		FAŁSZ	PRAWDA	Czy tramwaj jest niskopodłogowy
Liczba_kasownikow		Bigint		FAŁSZ	PRAWDA	Liczba kasowników w tramwaju

**Encja Zajezdnia** - istnieje wiele obiektów tej encji, definiuje zajeżdnię tramwajową.

Name	Domain	Data Type	Data Type Param 1	Primary Identifier	Mandatory	Description
Nr_zajezdni		SmallInt		PRAWDA	PRAWDA	Unikatowy numer zajezdni
Adres		VarChar(%p1)	200	FAŁSZ	PRAWDA	Adres zajezdni, pole segmentowe{miasto,ulica, numer lokalu, kod pocztowy, poczta}
Liczba_tramwajow		Integer		FAŁSZ	PRAWDA	Maksymalna liczba tramwajów w zajeźdźni
Miejsca_parkingowe		Integer		FAŁSZ	PRAWDA	Liczba miejsc parkingowych przy zajeźdźni

**Encja Przystanek** - istnieje wiele obiektów tej encji, definiuje przystanek tramwajowy.

Name	Domain	Data Type	Data Type Param 1	Primary Identifier	Mandatory	Description
Nr_przystanku		SmallInt		PRAWDA	PRAWDA	Unikatowy numer przystanku
Nazwa_przystanku		VarChar(%p1)	50	FAŁSZ	PRAWDA	Nazwa przystanku
Adres		VarChar(%p1)	200	FAŁSZ	PRAWDA	Adres przystanku, pole segmentowe{miasto,ulica, numer lokalu, kod pocztowy, poczta}
Czy_biletomat		Boolean		FAŁSZ	PRAWDA	Czy na przystanku jest biletomat
Czy_wiata		Boolean		FAŁSZ	PRAWDA	Czy przystanek posiada wiatę
Rozkład_jazdy		VarChar(%p1)	1000	FAŁSZ	FAŁSZ	Rozkład jazdy, pole wielowartościowe, pole segmentowe <nr_linii, godzina przyjazdu>

**Encja Linia** - istnieje wiele obiektów tej encji, definiuje linie tramwajową.

Name	Domain	Data Type	Data Type Param 1	Primary Identifier	Mandatory	Description
Nr_linii		SmallInt		PRAWDA	PRAWDA	Unikatowy numer linii
Czy_nocna		Boolean		FAŁSZ	PRAWDA	Czy linia kursuje w nocy(22-6)
Liczba_przystankow		SmallInt		FAŁSZ	PRAWDA	Liczba przystanków na linii

### 3.2. Ustalenie związków między encjami i ich typów

#### — ZTM - Pracownik

ZTM może zatrudniać wielu pracowników, ale na początku działalności może nie zatrudniać żadnego. Każdy pracownik zatrudniany jest przez biuro ZTM.

#### — ZTM - Tramwaj

ZTM posiada tramwaje, może posiadać ich wiele, lub też żadnego na początku działalności. Każdy tramwaj należy do ZTM.

#### — ZTM - Zajeżdźnia

ZTM może posiadać wiele zajeżdźni lub na początku działania zero. Każda zajeżdźnia należy do ZTM.

#### — ZTM - Linia

ZTM może obsługiwać wiele linii, lub zero na początku działalności. Każda linia jest obsługiwana przez ZTM.

#### — ZTM - Przystanek

ZTM może posiadać wiele przystanków, lub zero na początku działalności. Każdy przystanek podlega pod ZTM.

#### — Tramwaj - Linia

Tramwaj może kursować na wielu liniach, bądź na żadnej jeżeli nie jest aktualnie używany. Linia może być obsługiwana przez wiele tramwajów lub przez żaden, jeżeli jest aktualnie wyłączona z użytkowania.

— **Pracownik - Tramwaj**

Niektórzy z pracowników mogą opiekować się tramwajem. Tramwaj może posiadać wielu opiekunów ale może również ich nie mieć.

— **Zajezdnia - Linia**

Zajezdnia może być podłączona do wielu linii. Linia może posiadać wiele zajezdni, obowiązkowo przynajmniej jedną.

— **Linia - Przystanek**

Linia może posiadać wiele przystanków, co najmniej jeden. Przystanek może wchodzić w skład wielu linii, ale może nie być w żadnej.

— **Zajezdnia - Tramwaj**

Tramwaj może być przechowywany w wielu zajezdniach podczas swojej posługi. Zajezdnia nie zawsze przechowuje tramwaje.

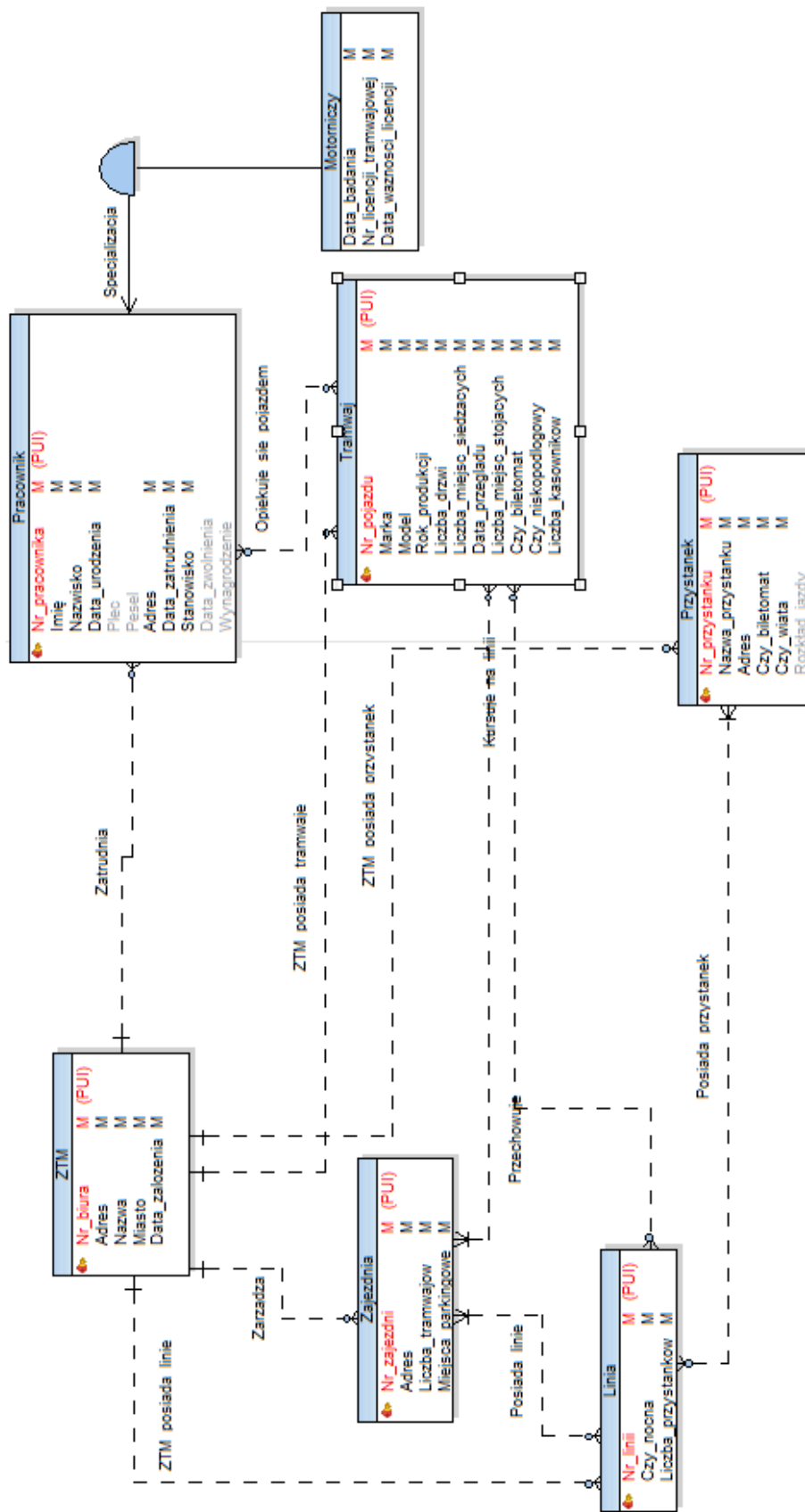
Relacja		Typ związku	Typ uczestnictwa	
ZTM	Pracownik	1:n	obowiązkowy	opcjonalny
ZTM	Tramwaj	1:n	obowiązkowy	opcjonalny
ZTM	Zajezdnia	1:n	obowiązkowy	opcjonalny
ZTM	Linia	1:n	obowiązkowy	opcjonalny
ZTM	Przystanek	1:n	obowiązkowy	opcjonalny
Tramwaj	Linia	n:m	opcjonalny	opcjonalny
Pracownik	Tramwaj	n:m	opcjonalny	opcjonalny
Zajezdnia	Linia	n:m	obowiązkowy	opcjonalny
Linia	Przystanek	n:m	opcjonalny	obowiązkowy
Zajezdnia	Tramwaj	n:m	obowiązkowy	opcjonalny

### 3.3. Klucze kandydujące i główne (decyzje projektowe)

Zdecydowaliśmy się na użycie jako kluczy sztuczne wygenerowane numery. Takie podejście zapewnia jednoznaczność danych oraz zwiększa czytelność bazy.

Nazwa encji	Klucz główny	Klucz kandydujący
ZTM	Nr_biura	Nazwa
Pracownik	Nr_pracownika	Pesel
Tramwaj	Nr_pojazdu	-
Zajezdnia	Nr_zajezdni	-
Linie	Nr_linii	-
Przystanek	Nr_przystanku	Nazwa_przystanku

### 3.4. Schemat ER na poziomie konceptualnym



### 3.5. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

#### 3.5.1. Przykład pułapki szczelinowej

Nie było możliwości utworzenia przystanku bez wpierw zdefiniowanej linii (w rzeczywistości linie tworzone są już na podstawie istniejących przystanków, ewentualnie dobudowują się nowe). Rozwiązaniem tego problemu było dodanie relacji między encjami ZTM oraz przystanku. Dzięki temu rozwiązaniu można stworzyć przystanek który podlega pod biuro ZTM i w późniejszym czasie dołączyć go do linii.

#### 3.5.2. Przykład pułapki wachlarzowej

Encja ZTM jest powiązana z encją Pracownik związkiem Zatrudnia. Ponadto encja ZTM posiada Tramwaje. Mimo, że mamy pracowników oraz tramwaje, to nie mogliśmy stwierdzić, który z pracowników (dokładniej motorniczych) użytkuje/opiekuje się danym tramwajem. Dlatego stworzyliśmy związek *Opiekuje się pojazdem* łączący pracownika z tramwajem.

## 4. Model logiczny

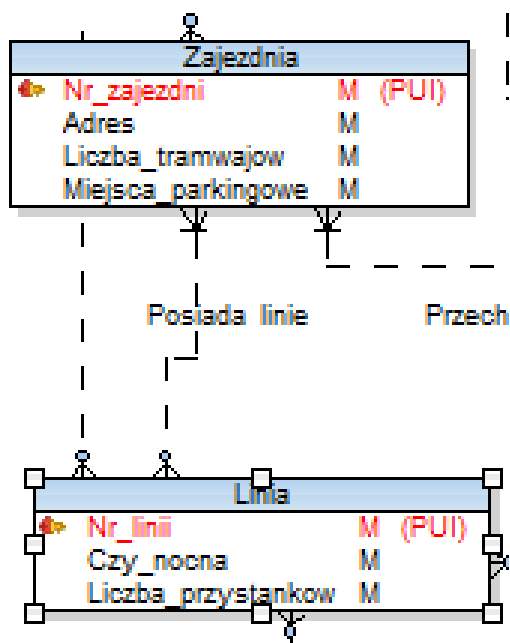
### 4.1. Charakterystyka modelu relacyjnego

Wygenerowano model relacyjny przy pomocy jednej z funkcji programu Toad Data Modeler 7.3. Aby wszystko działało poprawnie usunięto niekompatybilności z modelem koncepcyjnym (pokazane w sekcji poniżej) m.in. związki wiele do wielu. Celem zachowania unikatowości modelu koncepcyjnego zmieniono nazwy encji na liczby mnogie. Kolejnym krokiem było usunięcie z encji pól segmentowych poprzez stworzenie nowych encji na podstawie tych atrybutów. Zdefiniowano dodatkowo dwie dziedziny tj. dziedzinę godzin i minut. Ponadto pozbyto się specjalizacji i zastąpiono ją relacją.

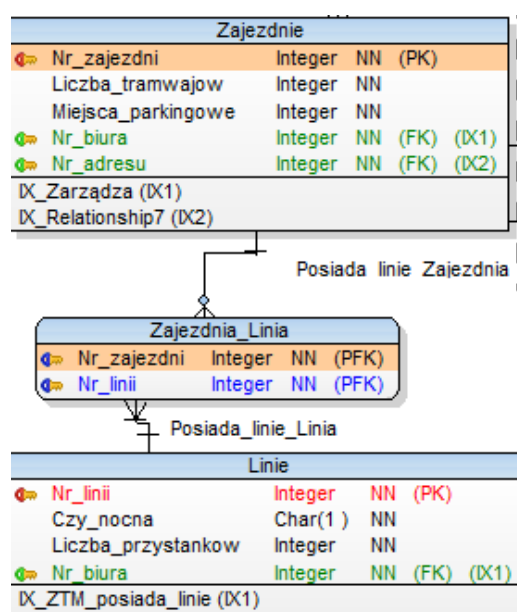
### 4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

#### 4.2.1. Usunięcie związków wiele do wielu

Wszystkie związki wiele do wielu (n:m) zostały wyeliminowane przy użyciu tablic bridge'ujących w których znajdują się klucze z obu encji. Poniżej przedstawiono przykład dla encji zajezdnie oraz linia.



Rysunek 1. Połączenie n:m encji zajezdnie i encji linia

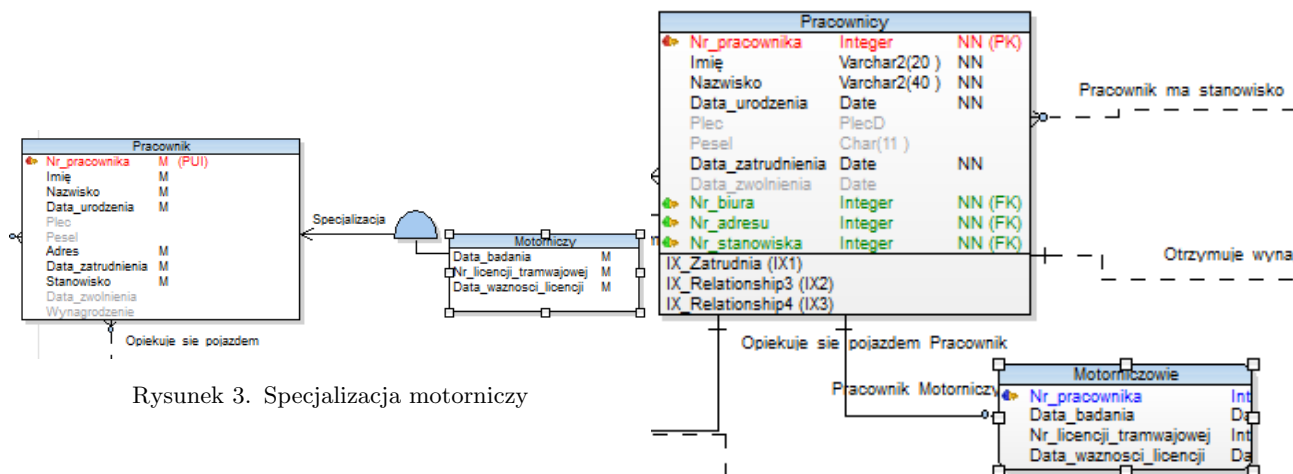


Rysunek 2. Tablica bridge'ująca powstała pomiędzy tablicami zajezdnie i encja



#### 4.2.2. Usunięcie specjalizacji

Usunięto specjalizację motorniczy i stworzono nową encję Motorniczowie, którą połączono z encją Pracownicy.



Rysunek 3. Specjalizacja motorniczy

Rysunek 4. Relacja Pracownicy - Motorniczowie

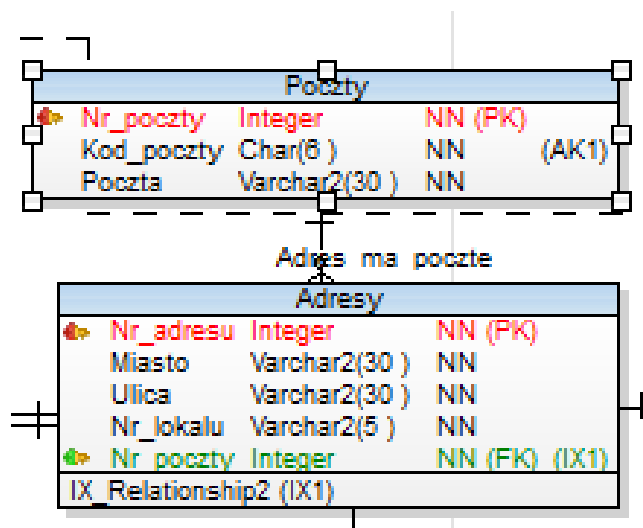
#### 4.3. Proces normalizacji – analiza i przykłady

##### 4.3.1. 1 postać normalna

1 postać normalna jest osiągnięta, jeżeli spełnione są następujące warunki:

- Wartości atrybutów są atomowe
- Nie zawiera powtarzających się grup informacji

Jednym z przykładów normalizacji do 1 postaci normalnej jest zastąpienie segmentowego atrybutu adres na nową encję Adresy, z której dalej wyklaruje się encja Poczty.



Rysunek 5. Encje Adresy i Poczty powstałe po normalizacji

##### 4.3.2. 2 postać normalna

2 postać normalna jest osiągnięta, jeżeli są spełnione następujące wymagania:

- Spełniona jest 1PN
- Żaden atrybut niekluczowy nie jest zależny od jakiegokolwiek klucza potencjalnego

W naszej bazie wszystkie rekordy zależą od klucza głównego więc baza spełnia kryteria 2 postaci normalnej

#### **4.3.3. 3 postać normalna**

3 postać normalna jest osiągnięta, jeżeli są spełnione następujące wymagania:

- Spełniona jest 2PN
- Żaden atrybut niekluczowy nie jest zależny od jakiegokolwiek atrybutu niekluczowego

Dobrym przykładem jest tabela Stanowiska, która umożliwia stworzenie bazy stanowisk potrzebnych jako informacja o pracowniku.

#### **4.4. Schemat ER na poziomie modelu logicznego**



#### 4.5. Więzy integralności

W tym momencie wszystkie pola segmentowe zostały poddane takim operacjom, że występują tylko pola atomowe. Duża część atrybutów posiada zaznaczone wartości NOT NULL. Wszystkie klucze główne są unikalne w swojej dziedzinie.

### 5. Faza fizyczna

#### 5.1. Projekt transakcji i weryfikacja ich wykonywalności

Transakcja	Potrzebne zasoby	Wykonalne?
Podgląd danych ZTM	ZTM i Adresy	Tak
Dodawanie, usuwanie, modyfikacja pracownika i jego danych	Pracownicy, adresy, stanowiska	Tak
Podgląd danych pracownika	Pracownicy, adresy, stanowiska	Tak
Dodawanie, usuwanie, modyfikacja tramwaju i jego danych	Tramwaje, Modele, Marki	Tak
Podgląd danych tramwaju	Tramwaje, Modele, Marki	Tak
Dodawanie, usuwanie, modyfikacja przystanku i jego danych	Przystanki, adres	Tak
Podgląd danych przystanku	Przystanki, adres	Tak
Dodawanie, usuwanie, modyfikacja zajezdni i jej danych	Zajezdnie, adres	Tak
Podgląd danych zajezdni	Zajezdnie, adres	Tak
Dodawanie, usuwanie, modyfikacja linii i jej danych	Linie	Tak
Podgląd danych linii	Linie	Tak
Dodawanie, usuwanie, modyfikacja stanowiska i jego danych	Stanowiska	Tak
Podgląd danych stanowiska	Stanowiska	Tak
Dodawanie, usuwanie, modyfikacja marki tramwaju i jej danych	Modele, Marki	Tak
Podgląd danych marki tramwaju	Modele, Marki	Tak
Dodawanie, usuwanie, modyfikacja modelu tramwaju i jego danych	Modele, Marki	Tak
Podgląd danych modelu tramwaju	Modele, Marki	Tak
Dodawanie, usuwanie, modyfikacja rozkładu jazdy i jego danych	Godziny przyjazdów, Linie, Przystanki	Tak
Podgląd danych rozkładu jazdy	Godziny przyjazdów, Linie, Przystanki	Tak
Podgląd listy wynagrodzeń	Wynagrodzenia	Tak

#### 5.2. Strojenie bazy danych – dobór indeksów

Poniżej przedstawiono wygenerowane indeksy dla poszczególnych tabel.

Indeksy dla ZTM:

```
CREATE INDEX IX_Relationship1 ON ZTM (Nr_adresu)
```

Indeksy dla Pracownicy:

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

```
CREATE INDEX IX_Relationship3 ON Pracownicy (Nr_adresu)
```

```
CREATE INDEX IX_Relationship4 ON Pracownicy (Nr_stanowiska)
```

Indeksy dla Tramwaje:

```
CREATE INDEX IX_ZTM_posiada_tramwaje ON Tramwaje (Nr_biura)
CREATE INDEX IX_Relationship12 ON Tramwaje (Nr_modelu)
```

Indeksy dla Zajezdnie:

```
CREATE INDEX IX_Zarzadza ON Zajezdnie (Nr_biura)
CREATE INDEX IX_Relationship7 ON Zajezdnie (Nr_adresu)
```

Indeksy dla Linie:

```
CREATE INDEX IX_ZTM_posiada_linie ON Linie (Nr_biura)
```

Indeksy dla Przystanki:

```
CREATE INDEX IX_ZTM_posiada_przystanek ON Przystanki (Nr_biura)
CREATE INDEX IX_Relationship8 ON Przystanki (Nr_adresu)
```

Indeksy dla Przyjazd\_na\_przystanek:

```
CREATE INDEX Godziny ON Przyjazd_na_przystanek (Nr_godziny)
```

Indeksy dla Adresy:

```
CREATE INDEX IX_Relationship2 ON Adresy (Nr_poczty)
```

Indeksy dla Wynagrodzenia:

```
CREATE INDEX IX_Relationship6 ON Wynagrodzenia (Nr_pracownika)
```

Indeksy dla Modele:

```
CREATE INDEX IX_Relationship9 ON Modele (Nr_marki)
```

### 5.3. Skrypt SQL zakładający bazę danych

Skrypt SQL zakładający bazę danych został wygenerowany przy użyciu jednej z funkcji programu Toad Data Modeler for Oracle

```
/*
Created: 13.11.2020
Modified: 19.11.2020
Model: Logical model
Database: Oracle 19c
*/

----- Create sequences section -----

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

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

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

```

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

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

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

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

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

CREATE SEQUENCE Przystankiseq
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

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

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

CREATE SEQUENCE Markiseq
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20

```

```

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

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

— Create tables section —————

— Table ZTM

CREATE TABLE ZTM(
    Nr_biura Integer NOT NULL,
    Nazwa Varchar2(50 ) NOT NULL,
    Miasto Varchar2(50 ) NOT NULL,
    Data_zalozenia Date NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

— Create indexes for table ZTM

CREATE INDEX IX_Relationship1 ON ZTM (Nr_adresu)
/

— Add keys for table ZTM

ALTER TABLE ZTM ADD CONSTRAINT BiuroPK PRIMARY KEY (Nr_biura)
/

— Table Pracownicy

CREATE TABLE Pracownicy(
    Nr_pracownika Integer NOT NULL,
    Imię Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(40 ) NOT NULL,
    Data_urodzenia Date NOT NULL,
    Plec Char(1 )
        CHECK (Plec IN ( 'K', 'M' ) ),
    Pesel Char(11 ),
    Data_zatrudnienia Date NOT NULL,
    Data_zwolnienia Date,
    Nr_biura Integer NOT NULL,
    Nr_adresu Integer NOT NULL,
    Nr_stanowiska Integer NOT NULL
)
/

— Create indexes for table Pracownicy

CREATE INDEX IX_Zatrudnia ON Pracownicy (Nr_biura)
/

CREATE INDEX IX_Relationship3 ON Pracownicy (Nr_adresu)
/

CREATE INDEX IX_Relationship4 ON Pracownicy (Nr_stanowiska)
/

```

```

— Add keys for table Pracownicy

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

— Table Motorniczowie

CREATE TABLE Motorniczowie(
    Nr_pracownika Integer NOT NULL,
    Data_badania Date NOT NULL,
    Nr_licencji_tramwajowej Integer NOT NULL,
    Data_waznosci_licencji Date NOT NULL
)
/

— Add keys for table Motorniczowie

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

— Table Tramwaje

CREATE TABLE Tramwaje(
    Nr_pojazdu Integer NOT NULL,
    Rok_produkcji Integer NOT NULL,
    Liczba_drzwi Integer NOT NULL,
    Liczba_miejsc_siedzacych Integer NOT NULL,
    Data_przeglądu Date NOT NULL,
    Liczba_miejsc_stojacych Integer NOT NULL,
    Czy_biletomat Char(1 ) NOT NULL,
    Czy_niskopodlogowy Integer NOT NULL,
    Liczba_kasownikow Integer NOT NULL,
    Nr_biura Integer NOT NULL,
    Nr_modelu Integer NOT NULL
)
/

— Create indexes for table Tramwaje

CREATE INDEX IX_ZTM_posiada_tramwaje ON Tramwaje (Nr_biura)
/

CREATE INDEX IX_Relationship12 ON Tramwaje (Nr_modelu)
/

— Add keys for table Tramwaje

ALTER TABLE Tramwaje ADD CONSTRAINT Unique_Identifier6 PRIMARY KEY (Nr_pojazdu)
/

— Table Zajezdnie

CREATE TABLE Zajezdnie(
    Nr_zajezdni Integer NOT NULL,
    Liczba_tramwajow Integer NOT NULL,
    Miejsca_parkingowe Integer NOT NULL,
    Nr_biura Integer NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

— Create indexes for table Zajezdnie

CREATE INDEX IX_Zarzadza ON Zajezdnie (Nr_biura)
/

CREATE INDEX IX_Relationship7 ON Zajezdnie (Nr_adresu)
/

— Add keys for table Zajezdnie

```



```

ALTER TABLE Zajezdnie ADD CONSTRAINT Unique_Identifier9 PRIMARY KEY (Nr_zajezdni)
/

— Table Linie

CREATE TABLE Linie(
    Nr_linii Integer NOT NULL,
    Czy_nocna Char(1 ) NOT NULL,
    Liczba_przystankow Integer NOT NULL,
    Nr_biura Integer NOT NULL
)
/

— Create indexes for table Linie

CREATE INDEX IX_ZTM_posiada_linie ON Linie (Nr_biura)
/

— Add keys for table Linie

ALTER TABLE Linie ADD CONSTRAINT Unique_Identifier10 PRIMARY KEY (Nr_linii)
/

— Table Przystanki

CREATE TABLE Przystanki(
    Nr_przystanku Integer NOT NULL,
    Nazwa_przystanku Varchar2(50 ) NOT NULL,
    Czy_biletomat Char(1 ) NOT NULL,
    Czy_wiata Char(1 ) NOT NULL,
    Nr_biura Integer NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

— Create indexes for table Przystanki

CREATE INDEX IX_ZTM_posiada_przystanek ON Przystanki (Nr_biura)
/

CREATE INDEX IX_Relationship8 ON Przystanki (Nr_adresu)
/

— Add keys for table Przystanki

ALTER TABLE Przystanki ADD CONSTRAINT Unique_Identifier11 PRIMARY KEY (Nr_przystanku)
/

— Table Zajezdnia_Linia

CREATE TABLE Zajezdnia_Linia(
    Nr_zajezdni Integer NOT NULL,
    Nr_linii Integer NOT NULL
)
/

— Table Zajezdnia_Tramwaj

CREATE TABLE Zajezdnia_Tramwaj(
    Nr_zajezdni Integer NOT NULL,
    Nr_pojazdu Integer NOT NULL
)
/

— Table Przyjazd_na_przystanek

CREATE TABLE Przyjazd_na_przystanek(
    Nr_linii Integer NOT NULL,
    Nr_przystanku Integer NOT NULL,

```

```

    Nr_godziny Integer NOT NULL
)
/

-- Create indexes for table Przyjazd_na_przystanek

CREATE INDEX Godziny ON Przyjazd_na_przystanek (Nr_godziny)
/

-- Table Pracownik_Tramwaj

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

-- Table Tramwaj_Linia

CREATE TABLE Tramwaj_Linia(
    Nr_pojazdu Integer NOT NULL,
    Nr_linii Integer NOT NULL
)
/

-- Table Adresy

CREATE TABLE Adresy(
    Nr_adresu Integer NOT NULL,
    Miasto Varchar2(30 ) NOT NULL,
    Ulica Varchar2(30 ) NOT NULL,
    Nr_lokalu Varchar2(5 ) NOT NULL,
    Nr_poczty Integer NOT NULL
)
/

-- Create indexes for table Adresy

CREATE INDEX IX_Relationship2 ON Adresy (Nr_poczty)
/

-- Add keys for table Adresy

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

-- Table and Columns comments section

COMMENT ON COLUMN Adresy.Nr_adresu IS 'Unikatowy identyfikator adresu'
/
COMMENT ON COLUMN Adresy.Miasto IS 'Miasto'
/
COMMENT ON COLUMN Adresy.Ulica IS 'Ulica'
/
COMMENT ON COLUMN Adresy.Nr_lokalu IS 'Numer lokalu'
/

-- Table Poczty

CREATE TABLE Poczty(
    Nr_poczty Integer NOT NULL,
    Kod_poczty Char(6 ) NOT NULL,
    Poczta Varchar2(30 ) NOT NULL
)
/

-- Add keys for table Poczty

ALTER TABLE Poczty ADD CONSTRAINT PK_Poczty PRIMARY KEY (Nr_poczty)
/

```

```

ALTER TABLE Poczty ADD CONSTRAINT Kod_poczty UNIQUE (Kod_poczty)
/

— Table and Columns comments section

COMMENT ON COLUMN Poczty.Nr_poczty IS 'Identyfikator poczty'
/
COMMENT ON COLUMN Poczty.Kod_poczty IS 'Kod poczty'
/
COMMENT ON COLUMN Poczty.Poczta IS 'Lokalizacja poczty'
/

— Table Stanowiska

CREATE TABLE Stanowiska(
    Nr_stanowiska Integer NOT NULL,
    Nazwa Varchar2(30 ) NOT NULL,
    Opis Varchar2(400 ) NOT NULL
)
/

— Add keys for table Stanowiska

ALTER TABLE Stanowiska ADD CONSTRAINT PK_Stalowiska PRIMARY KEY (Nr_stanowiska)
/

ALTER TABLE Stanowiska ADD CONSTRAINT Nazwa_stanowiska UNIQUE (Nazwa)
/

— Table and Columns comments section

COMMENT ON COLUMN Stanowiska.Nr_stanowiska IS 'Unikatowy identyfikator stanowiska'
/
COMMENT ON COLUMN Stanowiska.Nazwa IS 'Nazwa stanowiska'
/
COMMENT ON COLUMN Stanowiska.Opis IS 'Opis stanowiska '
/

— Table Wynagrodzenia

CREATE TABLE Wynagrodzenia(
    Nr_wynagrodzenia Integer NOT NULL,
    Data 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_Relationship6 ON Wynagrodzenia (Nr_pracownika)
/

— Add keys for table Wynagrodzenia

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

— Table and Columns comments section

COMMENT ON COLUMN Wynagrodzenia.Nr_wynagrodzenia IS 'Unikatowy numer wynagrodzenia'
/
COMMENT ON COLUMN Wynagrodzenia.Data IS 'Data wypłaty wynagrodzenia'
/
COMMENT ON COLUMN Wynagrodzenia.Kwota_podstawowa IS 'Kwota podstawowa'
/
COMMENT ON COLUMN Wynagrodzenia.Kwota_dodatkowa IS 'Kwota dodatkowa'
/

```

```

— Table Marki

CREATE TABLE Marki(
    Nr_marki Integer NOT NULL,
    Nazwa Varchar2(20 ) NOT NULL,
    Opis Varchar2(400 )
)
/

— Add keys for table Marki

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

ALTER TABLE Marki ADD CONSTRAINT Nazwa UNIQUE (Nazwa)
/

— Table and Columns comments section

COMMENT ON COLUMN Marki.Nr_marki IS 'Unikatowy numer marki'
/
COMMENT ON COLUMN Marki.Nazwa IS 'Nazwa marki'
/
COMMENT ON COLUMN Marki.Opis IS 'Opis marki'
/

— Table Modele

CREATE TABLE Modele(
    Nr_modelu Integer NOT NULL,
    Kod_modelu Varchar2(15 ) NOT NULL,
    Opis Varchar2(400 ),
    Nr_marki Integer NOT NULL
)
/

— Create indexes for table Modele

CREATE INDEX IX_Relationship9 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 Kod_modelu UNIQUE (Kod_modelu)
/

— Table and Columns comments section

COMMENT ON COLUMN Modele.Nr_modelu IS 'Unikatowy identyfikator modelu'
/
COMMENT ON COLUMN Modele.Kod_modelu IS 'Kod modelu'
/
COMMENT ON COLUMN Modele.Opis IS 'Opis modelu'
/

— Table Godziny_przyjazdów

CREATE TABLE Godziny_przyjazdów(
    Nr_godziny Integer NOT NULL,
    Godzina Integer NOT NULL
        CHECK (Godzina between 0 and 23),
    Minuta Integer NOT NULL
        CHECK (Minuta between 0 and 59)
)
/

```

```

— Add keys for table Godziny-przyjazdów

ALTER TABLE Godziny-przyjazdów ADD CONSTRAINT PK-Godziny-przyjazdów PRIMARY KEY (Nr-godziny)
/

— Table and Columns comments section

COMMENT ON COLUMN Godziny-przyjazdów.Nr-godziny IS 'Unikatowy numer godziny'
/
COMMENT ON COLUMN Godziny-przyjazdów.Godzina IS 'Godzina'
/
COMMENT ON COLUMN Godziny-przyjazdów.Minuta IS 'Minuta'
/

— Trigger for sequence ZTMseq for column Nr-biura in table ZTM —————
CREATE OR REPLACE TRIGGER ts_ZTM_ZTMseq BEFORE INSERT
ON ZTM FOR EACH ROW
BEGIN
    :new.Nr-biura := ZTMseq.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_ZTM_ZTMseq AFTER UPDATE OF Nr-biura
ON ZTM FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr-biura in table ZTM as
    it uses sequence.');
```

```

END;
/

— Trigger for sequence Pracownikseq for column Nr-pracownika in table Pracownicy —————
CREATE OR REPLACE TRIGGER ts_Pracownicy_Pracownikseq BEFORE INSERT
ON Pracownicy FOR EACH ROW
BEGIN
    :new.Nr-pracownika := Pracownikseq.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Pracownicy_Pracownikseq 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 Pojazdyseq for column Nr-pojazdu in table Tramwaje —————
CREATE OR REPLACE TRIGGER ts_Tramwaje_Pojazdyseq BEFORE INSERT
ON Tramwaje FOR EACH ROW
BEGIN
    :new.Nr-pojazdu := Pojazdyseq.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Tramwaje_Pojazdyseq AFTER UPDATE OF Nr-pojazdu
ON Tramwaje FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr-pojazdu in table Tramwaje
    as it uses sequence.');
```

```

END;
/

— Trigger for sequence Zajezdniaseq for column Nr-zajezdni in table Zajezdnie —————
CREATE OR REPLACE TRIGGER ts_Zajezdnie_Zajezdniaseq BEFORE INSERT
ON Zajezdnie FOR EACH ROW
BEGIN
    :new.Nr-zajezdni := Zajezdniaseq.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Zajezdnie_Zajezdniaseq AFTER UPDATE OF Nr-zajezdni
ON Zajezdnie FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr-zajezdni in table Zajezdnie
```

```

    as it uses sequence.');
```

**END;**  
/

— *Trigger for sequence Linieseq for column Nr\_linii in table Linie* —————

**CREATE OR REPLACE TRIGGER** ts\_Linie\_Linieseq **BEFORE INSERT**  
**ON** Linie **FOR EACH ROW**  
**BEGIN**  
    :new.Nr\_linii := Linieseq.nextval;  
**END;**  
/

**CREATE OR REPLACE TRIGGER** tsu\_Linie\_Linieseq **AFTER UPDATE** **OF** Nr\_linii  
**ON** Linie **FOR EACH ROW**  
**BEGIN**  
    RAISE\_APPLICATION\_ERROR(−20010,'Cannot update column Nr\_linii in table Linie  
    as it uses sequence.');

**END;**  
/

— *Trigger for sequence Przystankiseq for column Nr\_przystanku in table Przystanki* —————

**CREATE OR REPLACE TRIGGER** ts\_Przystanki\_Przystankiseq **BEFORE INSERT**  
**ON** Przystanki **FOR EACH ROW**  
**BEGIN**  
    :new.Nr\_przystanku := Przystankiseq.nextval;  
**END;**  
/

**CREATE OR REPLACE TRIGGER** tsu\_Przystanki\_Przystankiseq **AFTER UPDATE** **OF** Nr\_przystanku  
**ON** Przystanki **FOR EACH ROW**  
**BEGIN**  
    RAISE\_APPLICATION\_ERROR(−20010,'Cannot update column Nr\_przystanku in table Przystanki  
    as it uses sequence.');

**END;**  
/

— *Trigger for sequence Adresseq for column Nr\_adresu in table Adresy* —————

**CREATE OR REPLACE TRIGGER** ts\_Adresy\_Adresseq **BEFORE INSERT**  
**ON** Adresy **FOR EACH ROW**  
**BEGIN**  
    :new.Nr\_adresu := Adresseq.nextval;  
**END;**  
/

**CREATE OR REPLACE TRIGGER** tsu\_Adresy\_Adresseq **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 Pocztaseq for column Nr\_poczty in table Poczty* —————

**CREATE OR REPLACE TRIGGER** ts\_Poczty\_Pocztaseq **BEFORE INSERT**  
**ON** Poczty **FOR EACH ROW**  
**BEGIN**  
    :new.Nr\_poczty := Pocztaseq.nextval;  
**END;**  
/

**CREATE OR REPLACE TRIGGER** tsu\_Poczty\_Pocztaseq **AFTER UPDATE** **OF** Nr\_poczty  
**ON** Poczty **FOR EACH ROW**  
**BEGIN**  
    RAISE\_APPLICATION\_ERROR(−20010,'Cannot update column Nr\_poczty in table Poczty  
    as it uses sequence.');

**END;**  
/

— *Trigger for sequence Satnowiskoseq for column Nr\_stanowiska in table Stanowiska* —————

**CREATE OR REPLACE TRIGGER** ts\_Stanowiska\_Satnowiskoseq **BEFORE INSERT**  
**ON** Stanowiska **FOR EACH ROW**  
**BEGIN**  
    :new.Nr\_stanowiska := Satnowiskoseq.nextval;  
**END;**

```

/
CREATE OR REPLACE TRIGGER tsu_Stanowiska_Satnowiskoseq AFTER UPDATE OF Nr_stanowiska
ON Stanowiska FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_stanowiska in table Stanowiska
    as it uses sequence.');
```

END;

```

/

-- Trigger for sequence Wynagrodzenieseq for column Nr_wynagrodzenia in table Wynagrodzenia
CREATE OR REPLACE TRIGGER ts_Wynagrodzenia_Wynagrodzenieseq BEFORE INSERT
ON Wynagrodzenia FOR EACH ROW
BEGIN
    :new.Nr_wynagrodzenia := Wynagrodzenieseq.nextval;
END;
/

CREATE OR REPLACE TRIGGER tsu_Wynagrodzenia_Wynagrodzenieseq 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 Markiseq for column Nr_marki in table Marki
CREATE OR REPLACE TRIGGER ts_Marki_Markiseq BEFORE INSERT
ON Marki FOR EACH ROW
BEGIN
    :new.Nr_marki := Markiseq.nextval;
END;
/

CREATE OR REPLACE TRIGGER tsu_Marki_Markiseq 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 Modeleseq for column Nr_modelu in table Modele
CREATE OR REPLACE TRIGGER ts_Modele_Modeleseq BEFORE INSERT
ON Modele FOR EACH ROW
BEGIN
    :new.Nr_modelu := Modeleseq.nextval;
END;
/

CREATE OR REPLACE TRIGGER tsu_Modele_Modeleseq 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 Godzinaseq for column Nr_godziny in table Godziny-przyjazdów
CREATE OR REPLACE TRIGGER ts_Godziny-przyjazdów_Godzinaseq BEFORE INSERT
ON Godziny-przyjazdów FOR EACH ROW
BEGIN
    :new.Nr_godziny := Godzinaseq.nextval;
END;
/

CREATE OR REPLACE TRIGGER tsu_Godziny-przyjazdów_Godzinaseq AFTER UPDATE OF Nr_godziny
ON Godziny-przyjazdów FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_godziny in table
    Godziny-przyjazdów as it uses sequence.');
```

END;

```

/

```

— Create foreign keys (relationships) section —

```
ALTER TABLE Pracownicy ADD CONSTRAINT Zatrudnia FOREIGN KEY  
(Nr_biura) REFERENCES ZIM (Nr_biura)  
/
```

```
ALTER TABLE Zajezdnie ADD CONSTRAINT Zarządza FOREIGN KEY  
(Nr_biura) REFERENCES ZIM (Nr_biura)  
/
```

```
ALTER TABLE Tramwaje ADD CONSTRAINT ZTM_posiada_tramwaje FOREIGN KEY  
(Nr_biura) REFERENCES ZIM (Nr_biura)  
/
```

```
ALTER TABLE Linie ADD CONSTRAINT ZTM_posiada_linie FOREIGN KEY  
(Nr_biura) REFERENCES ZIM (Nr_biura)  
/
```

```
ALTER TABLE Przystanki ADD CONSTRAINT ZTM_posiada_przystanek FOREIGN KEY  
(Nr_biura) REFERENCES ZIM (Nr_biura)  
/
```

```
ALTER TABLE ZIM ADD CONSTRAINT BiuroZTM_ma_adres FOREIGN KEY  
(Nr_adresu) REFERENCES Adresy (Nr_adresu)  
/
```

```
ALTER TABLE Adresy ADD CONSTRAINT Adres_ma_poczte FOREIGN KEY  
(Nr_poczty) REFERENCES Poczty (Nr_poczty)  
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_ma_adres FOREIGN KEY  
(Nr_adresu) REFERENCES Adresy (Nr_adresu)  
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_ma_stanowisko FOREIGN KEY  
(Nr_stanowiska) REFERENCES Stanowiska (Nr_stanowiska)  
/
```

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

```
ALTER TABLE Zajezdnie ADD CONSTRAINT Zajezdnia_ma_adres FOREIGN KEY  
(Nr_adresu) REFERENCES Adresy (Nr_adresu)  
/
```

```
ALTER TABLE Przystanki ADD CONSTRAINT Przystanek_ma_adres FOREIGN KEY  
(Nr_adresu) REFERENCES Adresy (Nr_adresu)
```



/

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

/

```
ALTER TABLE Tramwaje ADD CONSTRAINT Tramwaj_ma_model FOREIGN KEY  
(Nr_modelu) REFERENCES Modele (Nr_modelu)
```

/

```
ALTER TABLE Przyjazd_na_przystanek ADD CONSTRAINT Przyjazd_godzina FOREIGN KEY  
(Nr_godziny) REFERENCES Godziny-przyjazdów (Nr_godziny)
```

/

#### 5.4. Skrypt SQL uzupełniający bazę o przykładowe dane

```
insert into MARKI(nazwa, opis) values ('PESA', 'Tramwaj marki PESA')
```

```
insert into MARKI(nazwa, opis) values ('CONSTAL', 'Tramwaj marki CONSTAL')
```

```
insert into MODELE(kod_modelu, opis, nr_marki) values  
( '116NA', 'Model tramwaju marki CONSTAL o kodzie 116NA', 2)
```

```
insert into MODELE(kod_modelu, opis, nr_marki) values  
( '2020N', 'Model tramwaju marki PESA o kodzie 2020N', 1)
```

```
insert into POCZTY(kod_poczty, poczta) values ('00-034', 'Warszawa')
```

```
insert into POCZTY(kod_poczty, poczta) values ('00-156', 'Warszawa')
```

```
insert into POCZTY(kod_poczty, poczta) values ('00-120', 'Warszawa')
```

```
insert into POCZTY(kod_poczty, poczta) values ('05-124', 'Skrzeszew')
```

```
insert into ADRESY(miasto, ulica, nr_lokalu, nr_poczty) values ('Warszawa', 'Nowogrodzka', '69', 1)
```

```
insert into ADRESY(miasto, ulica, nr_lokalu, nr_poczty) values ('Warszawa', 'Nowy Świat', '23', 3)
```

```
insert into ADRESY(miasto, ulica, nr_lokalu, nr_poczty) values ('Janówek  
Pierwszy', 'Nowodworska', '118A', 4)
```

```
insert into ADRESY(miasto, ulica, nr_lokalu, nr_poczty) values ('Warszawa', 'Złota',  
'1', 2)
```

```
insert into ADRESY(miasto, ulica, nr_lokalu, nr_poczty) values ('Warszawa', 'Mickiewicza', '420', 3)
```

```
insert into ADRESY(miasto, ulica, nr_lokalu, nr_poczty) values ('Warszawa', 'Mickiewicza', '32', 3)
```

```
insert into ADRESY(miasto, ulica, nr_lokalu, nr_poczty) values ('Warszawa', 'Ogrodowa', '12B', 1)
```

```
insert into ADRESY(miasto, ulica, nr_lokalu, nr_poczty) values  
( 'Warszawa', 'MaciusiaPierwszego', '21B', 1)
```

```
insert into ZTM(nazwa, miasto, data_zalozenia, nr_adresu) values  
( 'Tramwajowe przedsiębiorstwo miejskie', 'Warszawa', '19970128', 2)
```

```
insert into STANOWISKA(nazwa, opis) values ('Księgowa', 'Osoba odpowiedzialna za finanse')
```

```
insert into STANOWISKA(nazwa, opis) values ('Motorniczy', 'Osoba kierująca tramwajem')
```

```
insert into STANOWISKA(nazwa, opis) values ('Kontroler biletów',  
'Osoba odpowiedzialna za kontrolę biletów')
```

```
insert into STANOWISKA(nazwa, opis) values ('Logistyk',
```

```
'Osoba odpowiedzialna za logistykę np. rozkład jazdy')
```

```
insert into STANOWISKA(nazwa, opis) values ('Konserwator',  
'Osoba odpowiedzialna za dbanie o własność firmy')
```

```
insert into PRACOWNICY(imię, nazwisko, data_urodzenia, plec, pesel,
```

```

data_zatrudnienia , nr_biura ,nr_adresu ,nr_stanowiska)
values('Jan','Kowalski','19821105','M',98048651851,'20150901',1,3,2)
insert into PRACOWNICY(imię, nazwisko, data_urodzenia, plec, pesel,
data_zatrudnienia, nr_biura, nr_adresu, nr_stanowiska)
values ('Urszula','Dreszer','19861015','K',86041568131,'20160505',1,5,1)
insert into PRACOWNICY(imię, nazwisko, data_urodzenia, data_zatrudnienia,
nr_biura, nr_adresu, nr_stanowiska)
values('Sasza','Dmytro','19980101','20180719',1,4,5)
insert into PRACOWNICY(imię, nazwisko, data_urodzenia, plec,
data_zatrudnienia, data_zwolnienia, nr_biura, nr_adresu, nr_stanowiska)
values('Jakub','Dreszer','19850217','M','20120928','20170219',1,5,3)

insert into MOTORNICZOWIE(nr_pracownika, data_badania,
nr_licencji_tramwajowej, data_waznosci_licencji) values(2,'20200319',98415,'20220401')

insert into ZAJEZDNIE(liczba_tramwajow, miejsca_parkingowe, nr_biura, nr_adresu) values(20,200,1,1)
insert into ZAJEZDNIE(liczba_tramwajow, miejsca_parkingowe, nr_biura, nr_adresu) values(20,250,1,6)

insert into LINIE(czy_nocna, liczba_przystankow, nr_biura) values(1,14,1)

insert into PRZYSTANKI(nazwa_przystanku, czy_biletomat, czy_wiata, nr_biura, nr_adresu)
values('Bohaterów Modlina',1,1,1,7)
insert into PRZYSTANKI(nazwa_przystanku, czy_biletomat, czy_wiata, nr_biura, nr_adresu)
values('Rondo Maćka',0,1,1,8)

insert into TRAMWAJE(rok_produkcji, liczba_drzwi, liczba_miejsc_siedzacych, data_przeglądu,
liczba_miejsc_stojacych, czy_biletomat, czy_niskopodlogowy, liczba_kasownikow, nr_biura, nr_modelu)
values(2010,10,60,'20200120',30,1,1,2,1,1)
insert into TRAMWAJE(rok_produkcji, liczba_drzwi, liczba_miejsc_siedzacych, data_przeglądu,
liczba_miejsc_stojacych, czy_biletomat, czy_niskopodlogowy, liczba_kasownikow, nr_biura, nr_modelu)
values(2012,10,60,'20210501',30,0,1,2,1,2)

insert into WYNAGRODZENIA(data, kwota_podstawowa, nr_pracownika)
values('20200110', 3000,1)
insert into WYNAGRODZENIA(data, kwota_podstawowa, kwota_dodatkowa, nr_pracownika)
values('20200110', 3000,200,2)
insert into WYNAGRODZENIA(data, kwota_podstawowa, kwota_dodatkowa, nr_pracownika)
values('20200210', 3000,200,2)
insert into WYNAGRODZENIA(data, kwota_podstawowa, nr_pracownika) values('20200310', 3500,2)

insert into GODZINY_PRZYJAZDÓW(godzina, minuta) values(12,12)
insert into GODZINY_PRZYJAZDÓW(godzina, minuta) values(12,30)
insert into GODZINY_PRZYJAZDÓW(godzina, minuta) values(13,10)

insert into PRZYJAZD_NA_PRZYSTANEK(nr_linii, nr_przystanku, nr_godziny) values(1,1,1)
insert into PRZYJAZD_NA_PRZYSTANEK(nr_linii, nr_przystanku, nr_godziny) values(1,2,2)
insert into PRZYJAZD_NA_PRZYSTANEK(nr_linii, nr_przystanku, nr_godziny) values(1,1,3)

```

## 5.5. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

*Wypisanie rozkładu jazdy dla przystanku o nr 1*

```

select nr_linii, godzina, minuta
from PRZYJAZD_NA_PRZYSTANEK
inner join GODZINY_PRZYJAZDÓW
on PRZYJAZD_NA_PRZYSTANEK.nr_godziny = GODZINY_PRZYJAZDÓW.nr_godziny
where PRZYJAZD_NA_PRZYSTANEK.nr_przystanku=1;

```

*Wypisanie wszystkich pracowników pracujących jako księgowa*

```
select * from PRACOWNICY p
  inner join STANOWISKA s
    on p.nr_stanowiska=s.nr_stanowiska
 where s.NAZWA='Księgowa';
```

*Wypisanie pracowników mieszkających w Warszawie*

```
select * from PRACOWNICY p
  inner join ADRESY a
    on p.nr_adresu=a.nr_adresu
 where a.MIASTO = 'Warszawa';
```

*Sprawdzenie historii wypłat przez pracownika o imieniu Urszula Dreszer*

```
select w.* from WYNAGRODZENIA w
  inner join PRACOWNICY p
    on w.nr_pracownika=p.nr_pracownika
 where p.imię='Urszula' and p.nazwisko='Dreszer';
```

*Wyszukiwanie zwolnionych pracowników*

```
select * from PRACOWNICY
 where data_zwolnienia IS NOT NULL;
```

## 6. Bibliografia

- Wykłady z przedmiotu BDBT -dr. inż. M.Kowalczyk
- Nagrania z zajęć projektowych z przedmiotu BDBT -dr. inż. M.Kowalczyk
- Dokumentacja bazy danych Oracle: <https://docs.oracle.com/en/database/>