Miejskie Przedsiębiorstwo Tramwajowe

Dokumentacja Projektu

Jan Wojciechowski

Kacper Szaruch

2022

Spis treści

[Zakres i cel projektu (opis założeń funkcjonalnych projektowanej bazy danych) 2](#_Toc121697683)

[Definicja systemu 2](#_Toc121697684)

[Perspektywy użytkowników 3](#_Toc121697685)

[Model konceptualny 3](#_Toc121697686)

[Definicja zbiorów encji określonych w projekcie (decyzje projektowe) 3](#_Toc121697687)

[Ustalenie związków między encjami i ich typów 4](#_Toc121697688)

[Określenie atrybutów i ich dziedzin 5](#_Toc121697689)

[Dodatkowe reguły integralnościowe (reguły biznesowe) 9](#_Toc121697690)

[Klucze kandydujące i główne (decyzje projektowe) 10](#_Toc121697691)

[Schemat ER na poziomie konceptualnym 10](#_Toc121697692)

[Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady 10](#_Toc121697693)

[Model logiczny 11](#_Toc121697694)

[Charakterystyka modelu relacyjnego 11](#_Toc121697695)

[Usunięcie właściwości niekompatybilnych z modelem relacyjnym – przykłady 11](#_Toc121697696)

[Proces normalizacji – analiza i przykłady 11](#_Toc121697697)

[Schemat ER na poziomie modelu logicznego 13](#_Toc121697698)

[Więzy integralności 13](#_Toc121697699)

[Proces denormalizacji – analiza i przykłady 13](#_Toc121697700)

[Faza fizyczna 14](#_Toc121697701)

[Projekt transakcji i weryfikacja ich wykonalności 14](#_Toc121697702)

[Strojenie bazy danych – dobór indeksów 15](#_Toc121697703)

[Skrypt SQL zakładający bazę danych 15](#_Toc121697704)

[Przykłady zapytań i poleceń SQL odnoszących się do bazy danych 24](#_Toc121697705)

[Bibliografia 25](#_Toc121697706)

# Zakres i cel projektu (opis założeń funkcjonalnych projektowanej bazy danych)

Celem projektu było opracowanie oraz zaimplementowanie „relacyjnej bazy danych”.

Osiągnięcie tego zadania wymagało realizacji trzech faz składających się na cały

proces: konceptualnej, relacyjnej oraz fizycznej. Pierwsze dwie fazy dotyczyły modelowania

schematu bazy danych (model/schemat ER – Entity Relationship), natomiast trzecia –

właściwej implementacji w określonym silniku bazy danych.

Zrealizowana baza danych dotyczy obsługi i zarządzania przedsiębiorstwem tramwajowym. Informacje, które można w niej przechowywać dotyczą nie tylko zajezdni tramwajowych i ich możliwości serwisowych, ale również pracownikach zatrudnianych przez przedsiębiorstwo oraz kasowanych biletach krótko i długookresowych. Baza opisuje również umiejętności i specjalizacje poszczególnych pracowników (takie jak znajomość języków czy posiadane uprawnienia do prowadzenia pojazdami razem z datą ich ważności) oraz oferowane w tramwajach lub na przystankach przestrzenie reklamowe. Dostępne są również podstawowe informacje o klientach (w szczególności w przypadku wykupienia biletu długookresowego imiennego).

W trakcie realizacji projektu wykorzystywaliśmy następujące oprogramowanie:

* Lokalna baza danych Oracle Database 21c
* Toad Data Modeler w wersji 7.2
* Oprogramowanie Java

# Definicja systemu

Definicja systemu zakłada następujące funkcjonalności:

* Podgląd informacji o tramwajach,
* Dodawanie/modyfikowanie/usuwanie informacji o tramwajach,
* Podgląd informacji o zajezdniach,
* Dodawanie/modyfikowanie/usuwanie informacji o zajezdniach,
* Podgląd informacji o biurach,
* Dodawanie/modyfikowanie/usuwanie informacji o biurach,
* Podgląd informacji o pracownikach,
* Dodawanie/modyfikowanie/usuwanie informacji o pracownikach,
* Podgląd informacji o pasażerach,
* Podgląd informacji o pracownikach jako kierowcy,
* Dodawanie/modyfikowanie/usuwanie informacji o pracownikach jako kierowcy,
* Podgląd informacji o pracownikach jako o pracownikach obsługi klienta,
* Dodawanie/modyfikowanie/usuwanie informacji o pracownikach jako o pracownikach obsługi klienta,
* Podgląd informacji o pracownikach jako o kontrolerach,
* Dodawanie/modyfikowanie/usuwanie informacji o pracownikach jako o kontrolerach,
* Podgląd informacji o wynagrodzeniu pracowników,
* Dodawanie/modyfikowanie/usuwanie informacji o wynagrodzeniu pracowników,
* Podgląd informacji o językach obcych,
* Dodawanie/modyfikowanie/usuwanie informacji o językach obcych,
* Podgląd informacji o znajomości języków obcych przez pracowników obsługi klienta,
* Dodawanie/modyfikowanie/usuwanie informacji o znajomości języków obcych przez pracowników obsługi klienta,
* Dodawanie/modyfikowanie/usuwanie informacji o pasażerach,
* Podgląd informacji o biletach,
* Dodawanie/modyfikowanie/usuwanie informacji o biletach,
* Podgląd informacji o reklamach i miejscach reklamowych,
* Dodawanie/modyfikowanie/usuwanie informacji o reklamach i miejscach reklamowych,
* Podgląd informacji o tramwajach,
* Dodawanie/modyfikowanie/usuwanie informacji o tramwajach

## Perspektywy użytkowników

Głównymi użytkownikami projektowanego systemu mogą być:

* **Główny zarządca (administrator)** – dostęp i prawo do modyfikacji wszystkich danych w bazie,
* **Pasażer** – dostęp do informacji na temat udogodnień w tramwajach i biurach obsługi klienta,
* **Pracownik** – dostęp do danych osobowych oraz informacji o wynagrodzeniu (wszyscy pracownicy), dostęp do informacji dotyczących wyposażenie zajezdni i tramwajów (pracownicy obsługi), dostęp do informacji dotyczących pasażerów i biletów (kontrolerzy)

# Model konceptualny

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

W projekcie na poziomie konceptualnym wyróżniamy następujące encje:

* **Tramwaj** – główna encja zawierająca informacje dotyczące tramwaju,
* **Zajezdnia** – encja, która zawiera informacje dotyczące wyposażenia i działania zajezdni,
* **Biuro** – encja, która zawiera informacje dotyczące biura obsługi klienta,
* **Pracownik** – encja, która zawiera informacje ogólne na temat pracownika Miejskiego Przedsiębiorstwa Tramwajowego,
* **Kierowca** – encja uszczegóławiająca pracownika jako kierowcę,
* **Obsługa** – encja uszczegóławiająca pracownika jako pracownika obsługi,
* **Kontroler** – encja uszczegóławiająca pracownika jako kontrolera,
* **Reklama –** encja zawierająca informacje dotyczące eksponowanych na przystankach i w tramwajach reklam,
* **Bilet –** encja zawierająca informacje ogólne dotyczące biletu,
* **Pasażer –** encja zawierająca podstawowe informacje na temat pasażera tramwaju.

## Ustalenie związków między encjami i ich typów

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa relacji | Krotność | Typ uczestnictwa | Opis |
| Tramwaj - Zajezdnia | Wiele do wielu | Tramwaj jest opcjonalny dla zajezdni, zajezdnia jest opcjonalna dla tramwaju | Zajezdnia może nie obsługiwać żadnego tramwaju, tramwaj może nie być obsługiwany przez żadną zajezdnię |
| Tramwaj – Pracownik | Wiele do wielu | Pracownik jest opcjonalny dla tramwaju, tramwaj jest opcjonalny dla pracownika | Tramwaj może nie być obsługiwany przez żadnego pracownika, pracownik może nie obsługiwać żadnego tramwaju |
| Tramwaj – Reklama | Wiele do wielu | Tramwaj jest opcjonalny dla reklamy, reklama jest opcjonalna dla tramwaju | Reklama może nie znajdować się w tramwaju (na przystanku), a tramwaj może nie mieć reklam |
| Pasażer - Tramwaj | Jeden do wielu | Tramwaj jest obowiązkowy dla pasażera, pasażer jest opcjonalny dla tramwaju | Tramwaj może mieć wielu pasażerów lub 0, a pasażer, żeby być pasażerem musi mieć przynajmniej jeden tramwaj |
| Pasażer – Bilet | Jeden do wielu | Pasażer jest obowiązkowy dla biletu, bilet jest obowiązkowy dla pasażera | Pasażer może mieć wiele biletów, ale musi mieć przynajmniej jeden, a jeden bilet może mieć tylko jeden pasażer |
| Zajezdnia – Pracownik | Jeden do wielu | Zajezdnia nie jest obowiązkowa dla pracownika, pracownik nie jest obowiązkowy dla zajezdni | Pracownik może pracować w co najwyżej jednej zajezdni, w zajezdni nie musi pracować żaden pracownik, ale może wiele |
| Biuro – Pracownik | Jeden do wielu | Biuro nie jest obowiązkowe dla pracownika, pracownik nie jest obowiązkowy dla biura | Pracownik może pracować w co najwyżej jednym biurze, biuro może mieć wielu pracowników |

## Określenie atrybutów i ich dziedzin

**Tramwaj**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
| Nr\_tramwaju | Integer | Obowiązkowy | Numer tramwaju |
| Model | VarChar(20) | Obowiązkowy | Model tramwaju |
| Czy\_klimatyzacja | Boolean | Obowiązkowy | Czy tramwaj ma klimatyzację |
| Czy\_ogrzewanie | Boolean | Obowiązkowy | Czy tramwaj ma ogrzewanie |
| Czy\_sprawny | Boolean | Obowiązkowy | Czy tramwaj jest sprawny |
| Czy\_niskopodlogowy | Boolean | Obowiązkowy | Czy tramwaj jest niskopodłogowy |
| Data\_przegladu | Date | Obowiązkowy | Data ostatniego przeglądu technicznego |
| Pojemnosc | SmallInt | Obowiązkowy | Pojemność tramwaju |
| Ilosc\_miejsc\_siedzacych | SmallInt | Obowiązkowy | Ilość miejsc siedzących w tramwaju |
| Ilosc\_miejsc\_niepelnosprawnych | SmallInt | Obowiązkowy | Ilość miejsc przeznaczonych dla osób z trudnościami w poruszaniu się |
| Ilosc\_kasownikow | SmallInt | Obowiązkowy | Ilość kasowników w tramwaju |
| Czy\_biletomat | Boolean | Obowiązkowy | Czy tramwaj posiada biletomat |

**Zajezdnia**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
| Nr\_zajezdni | SmallInt | Obowiązkowy | Numer zajezdni tramwajowej |
| Czy\_funkcjonuje | Boolean | Obowiązkowy | Czy zajezdnia funkcjonuje |
| Adres | VarChar(400) | Obowiązkowy | Adres zajezdni |
| Pojemnosc | SmallInt | Obowiązkowy | Maksymalna liczba tramwajów, które może obsłużyć zajezdnia |
| Godziny otwarcia | VarChar(32) | Obowiązkowy | Godziny otwarcia zajezdni |
| Czy\_myjnia | Boolean | Obowiązkowy | Czy w zajezdni znajduje się myjnia |
| Firma\_ochroniarska | VarChar(40) | Nieobowiązkowy | Nazwa firmy ochroniarskiej ochraniającej zajezdnię |
| Ilosc\_tramwajow\_serwisowych | SmallInt | Obowiązkowy | Ilość tramwajów serwisowych dostępnych w zajezdni |
| Czy\_automatyczne\_zwrotnice | Boolean | Obowiązkowy | Czy zajezdnia jest wyposażona w automatyczne zwrotnice |
| Czy\_lakiernia | Boolean | Obowiązkowy | Czy zajezdnia posiada lakiernię |
| Czy\_tokarka\_podtorowa | Boolean | Obowiązkowy | Czy zajezdnia posiada tokarkę podtorową |

**Biuro**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
| Nr\_biura | SmallInt | Obowiązkowy | Numer biura |
| Adres | VarChar(400) | Obowiązkowy | Adres biura. Pole segmentowe, które  zawiera kraj, miasto, ulicę, numer budynku,  numer lokalu i kod pocztowy |
| Nazwa | VarChar(20) | Obowiązkowy | Nazwa biura |
| Czy\_ulatwienia\_dostepu | Boolean | Obowiązkowy | Czy biuro posiada ułatwienia dostępu dla ludzi z ograniczoną sprawnością ruchową |
| Ilosc\_stanowisk | Integer | Obowiązkowy | Ilość stanowisk do obsługi klientów |

**Pasażer**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
| Nr\_klienta | Integer | Obowiązkowy | Numer pasażera |
| Imie | VarChar(20) | Nieobowiązkowy | Imię pasażera |
| Nazwisko | VarChar(30) | Nieobowiązkowy | Nazwisko pasażera |

**Bilet**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
| Nr\_biletu | Integer | Obowiązkowy | Numer biletu |
| Czy\_ulgowy | Boolean | Obowiązkowy | Czy bilet jest ulgowy |
| Data\_zakupu | DateTime | Nieobowiązkowy | Data i czas zakupu |
| Data\_skasowania | DateTime | Nieobowiązkowy | Data i czas skasowania biletu |
| Wazny\_do | DateTime | Nieobowiązkowy | Data i czas ważności biletu |
| Obslugiwane\_strefy | StrefyD (1,2) | Obowiązkowy | Obsługiwane strefy w ramach danego biletu |
| Czas\_waznosci | Time | Obowiązkowy | Czas ważności biletu |
| Czy\_imienny | Boolean | Obowiązkowy | Czy bilet jest imienny |

**Reklama**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
| Nr\_reklamy | Integer | Obowiązkowy | Numer reklamy |
| Format | VarChar(20) | Obowiązkowy | Format reklamy |
| Czy\_podswietlenie | Boolean | Obowiązkowy | Czy reklama jest podświetlana |
| Data\_konca | Date | Obowiązkowy | Data zakończenia okresu prezentacji reklamy |
| Dlugosc | Time | Nieobowiązkowy | Długość trwania reklamy |
| Lokalizacja | VarChar(50) | Obowiązkowy | Lokalizacja ekspozycji reklamy |

**Pracownik**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
| Nr\_pracownika | Integer | Obowiązkowy | Numer pracownika |
| Imie | VarChar(20) | Obowiązkowy | Imię pracownika |
| Nazwisko | VarChar(30) | Obowiązkowy | Nazwisko pracownika |
| Data\_urodzenia | Date | Obowiązkowy | Data urodzenia pracownika |
| PESEL | Character(11) | Obowiązkowy | Numer PESEL pracownika |
| Nazwisko\_panienskie\_matki | VarChar(30) | Obowiązkowy | Nazwisko panieńskie matki pracownika |
| Dowod\_osobisty | Character(9) | Obowiązkowy | Seria i numer dowodu osobistego pracownika |
| Adres | VarChar(400) | Obowiązkowy | Adres pracownika. Pole segmentowe, które zawiera kraj, miasto, ulicę, numer budynku, numer lokalu i kod pocztowy |
| Czy\_karany | Boolean | Obowiązkowy | Czy pracownik był karany |
| Wykształcenie | VarChar(30) | Obowiązkowy | Wykształcenie pracownika |
| Nr\_konta | Character(26) | Obowiązkowy | Numer konta pracownika |
| Nr\_telefonu | Character(9) | Obowiązkowy | Numer telefonu pracownika |
| Email | VarChar(40) | Nieobowiązkowy | Adres email pracownika |
| Wynagrodzenie | Money | Obowiązkowy | Wynagrodzenie pracownika |
| Plec | PlecD (K, M, Inne) | Obowiązkowy | Płeć pracownika |

**Kierowca**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
| Nr\_prawa\_jazdy | Character(14) | Obowiązkowy | Numer prawa jazdy kierowcy |
| Data\_waznosci\_prawa\_jazdy | Date | Obowiązkowy | Data ważności prawa jazdy |
| Data\_waznosci\_badan | Date | Obowiązkowy | Data ważności badań lekarskich |

**Obsługa**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
| Stanowisko | VarChar(20) | Obowiązkowy | Stanowisko zajmowane przez pracownika obsługi |
| Jezyk | VarChar(60) | Obowiązkowy | Języki, którymi posługuje się pracownik obsługi wraz z poziomem znajomości. Pole segmentowe, które zawiera język obcy  oraz stopień jego znajomości |
| Dodatkowe\_kursy | VarChar(400) | Nieobowiązkowy | Dodatkowe kursy wyrobione przez pracownika obsługi |

**Kontroler**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
| Licencja | VarChar(30) | Obowiązkowy | Numer licencji kontrolera biletów |
| Liczba\_wystawionych\_mandatow | Integer | Obowiązkowy | Liczba mandatów wystawionych przez kontrolera |

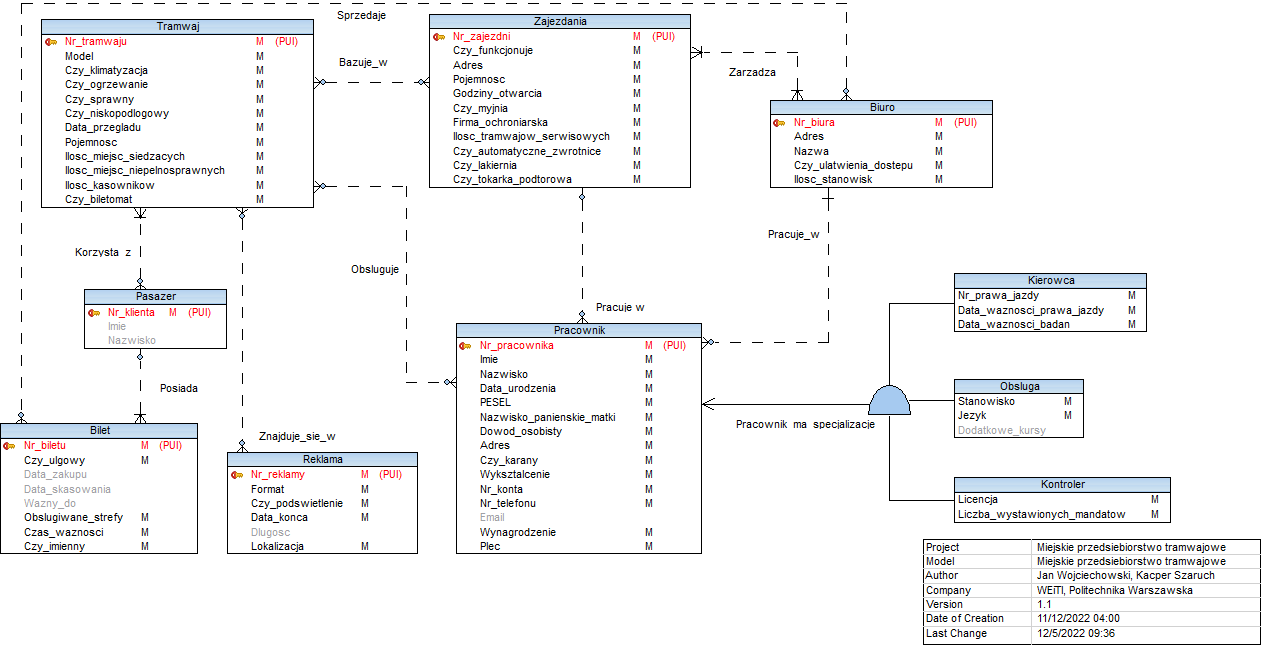
## Dodatkowe reguły integralnościowe (reguły biznesowe)

* Biuro może mieć ten sam adres co zajezdnia, jednak jest traktowane jako osobna jednostka
* Zajezdnie oraz tramwaje mogą być niefunkcjonujące, dlatego nie będzie do nich przypisanych żadnych pracowników

## Klucze kandydujące i główne (decyzje projektowe)

|  |  |  |
| --- | --- | --- |
| Nazwa encji | Klucz główny | Klucz Kandydujący |
| Tramwaj | Nr\_tramwaju | - |
| Zajezdnia | Nr\_zajezdni | Adres |
| Biuro | Nr\_biura | Adres, Nazwa |
| Pasazer | Nr\_klienta | - |
| Bilet | Nr\_biletu | - |
| Reklama | Nr\_reklamy | - |
| Pracownik | Nr\_pracownika | PESEL, Dowod\_osobisty, Adres, Nr\_konta, Nr\_telefonu |
| Kierowca | Nr\_pracownika | Nr\_prawa\_jazdy |
| Obsluga | Nr\_pracownika | - |
| Kontroler | Nr\_pracownika | Licencja |

## Schemat ER na poziomie konceptualnym



## Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

* **Pułapki wachlarzowe** – jest to niejednoznaczne powiązanie pomiędzy dwoma encjami, inaczej, oferuje wachlarz możliwości. W trakcie analizy nie znaleźliśmy pułapek wachlarzowych w naszym modelu. Jeżeli jednak taka pułapka by wystąpiła, najlepszym rozwiązaniem jest stworzenie doprecyzowanej relacji, która pozwoliłaby na dokładne określenie związku między dwoma encjami.
* **Pułapki szczelinowe** – w trakcie analizy schematu ER na poziomie konceptualnym oraz na poziomie modelu logicznego udało nam się zidentyfikować pułapkę szczelinową spowodowaną brakiem połączenia pomiędzy Biuro, a Bilet. Udało nam się naprawić ten problem dodając stosowne połączenie, które reprezentuje realną akcję sprzedaży biletów przez biuro.

# Model logiczny

## Charakterystyka modelu relacyjnego

Po zakończeniu etapu projektowania schematu ER bazy danych na poziomie konceptualnym

oraz weryfikacji jego poprawności, za pomocą odpowiedniej opcji w programie Toad Data

Modeler 7.2 dokonano procesu konwersji projektu do poziomu logicznego. Tym sposobem:

* każda relacja typu „wiele do wielu” zastąpiona została dwiema relacjami typu „jeden

do wielu” wraz z tabelą łączącą obie nowo powstałe relacje; dodatkowo – nowe relacje

po obu stronach charakteryzują się obowiązkowym typem uczestnictwa,

* tabele otrzymały dodatkowe atrybuty w postaci kluczy obcych oznaczonych na zielono,
* typy danych zostały przekształcone do tych używanych przez wybrany w procesie

konwersji silnik bazy danych,

* identyfikujący atrybut każdej encji stał się kluczem głównym tabeli,
* encje uszczegóławiające zostały przekształcone w tabele i uzupełnione o klucz główny

encji nadrzędnej oznaczony na niebiesko,

* pomiędzy tabelami uszczegóławiającymi a tabelami nadrzędnymi domyślnie stworzone

zostały relacje jeden do wielu z nieobowiązkowym typem uczestnictwa po stronie

potomka oraz obowiązkowym po stronie rodzica.

Dodatkowo, po konwersji:

* Usunęliśmy pola segmentowe – zamiast tego stworzyliśmy oddzielne tabele, tak aby spełniać założenia pierwszej postaci normalnej
* Zmieniliśmy nazwy tabel na liczbę mnogą

## Usunięcie właściwości niekompatybilnych z modelem relacyjnym – przykłady

Usunięcie właściwości niekompatybilnych z modelem relacyjnym objęło usunięcie pól wielowartościowych np. Adres – miasto, ulica, nr. Lokalu, kod pocztowy. Kolejnym krokiem było usunięcie związków wiele do wielu; pomiędzy tymi tabelami została utworzona tabela:

* Tramwaje – Zajezdnie; Miejsca\_postojowe
* Zajezdnie – Biura; Obsługi\_zajezdni
* Pasazerowie – Tramwaje; Pasazerowie\_tramwajow
* Tramwaje – Reklamy; Gabloty\_reklamowe
* Tramwaje – Pracownicy; Obslugi\_tramwajow

## Proces normalizacji – analiza i przykłady

Proces normalizacji bazy danych polega na separacji powtarzających się danych i przechowywaniu ich w jednym miejscu. Służy to poprawie bezpieczeństwa przechowywanych danych oraz przyspieszeniu i ułatwieniu aktualizacji danych. Zmniejsza się również ryzyko powstawania niespójności i anomalii. Wadą doprowadzenia bazy danych do kolejnych postaci normalnych jest możliwość skomplikowania schematu danych, a co za tym idzie, spowolnienie odczytu danych.

Pierwsza postać normalna jest osiągnięta, kiedy relacja:

* Opisuje jeden obiekt
* Wartości atrybutów są elementarne (atomowe)
* Nie zawiera kolekcji
* Kolejność wierszy może być dowolna

Sprowadzenie do pierwszej postaci normalnej w przypadku tego projektu polegało na wydzieleniu informacji o adresie i znajomości języków pracowników obsługi.

Atrybut „Adres” był polem segmentowym, ponieważ zawierał informację o mieście, ulicy itd.. Wyodrębnienie go do osobnej relacji, w której każdy segment można przechowywać osobno umożliwiło sprowadzenie go do pierwszej postaci normalnej.

Text, table

Description automatically generated

Kolejnym atrybutem, który posiadał podobny problem jest „Jezyk”. Było to pole wielowartościowe, ponieważ niezbędne jest określenie zarówno znanego języka jak i poziomu znajomości. Ponadto było to pole wielowartościowe, ponieważ pracownik obsługi mógł znać parę języków. Aby rozwiązać ten problem utworzyliśmy tabelę łączącą relacje „Jezyki”, „Poziomy\_znajomosci” i „Obslugi”.

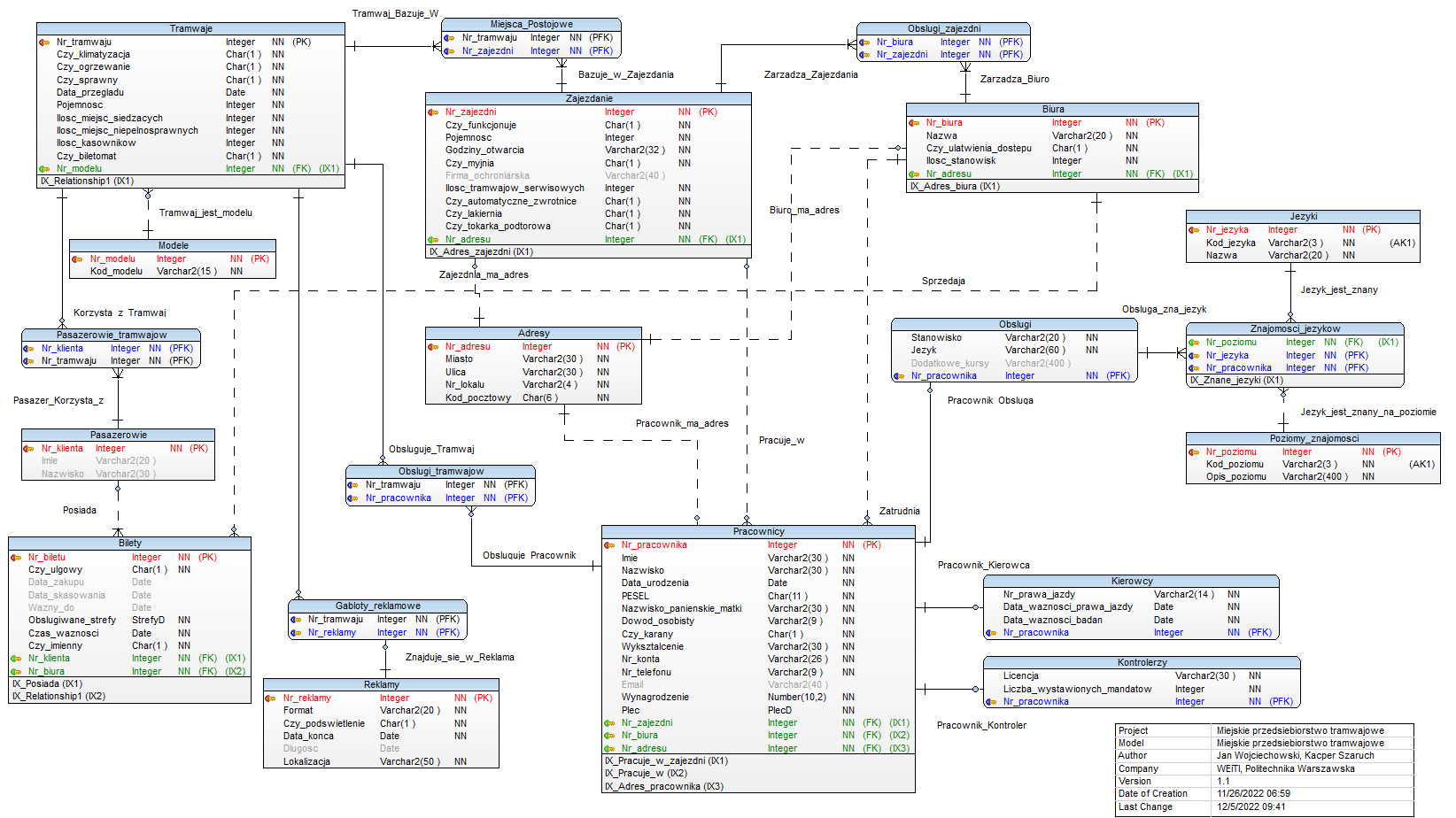
Diagram

Description automatically generated

Druga postać normalna została osiągnięta poprzez spełnienie wymagań pierwszej postaci oraz zastosowanie sztucznych kluczy Nr w każdej relacji. Gwarantuje to, że żadna kolumna niekluczowa nie jest częściowo funkcyjnie zależna od jakiegokolwiek klucza potencjalnego.

Trzecia postać normalna wymaga spełnienia założeń drugiej postaci normalnej oraz zapewnienia sytuacji, w której każdy atrybut danej relacji nie wchodzący w skład żadnego klucza potencjalnego nie był przechodnio funkcyjnie zależny od żadnego klucza potencjalnego tej relacji, a więc był określony tylko i wyłącznie kluczem. Ta postać została spełniona automatycznie, dzięki zastosowaniu kluczy głównych oraz upewnieniu się, że cały model został jak najbardziej uproszczony, tak aby precyzyjnie opisywać określony obiekt.

## Schemat ER na poziomie modelu logicznego



## Więzy integralności

Integralność danych jest osiągana poprzez zgodność typu przechowywanych danych z ich rzeczywistym charakterem. Jest to przez nas osiągane poprzez definiowanie dziedzin oraz dopuszczalnych formatów danych. Ponadto większość argumentów jest obowiązkowa, co powoduje, że tylko w niewielu przypadkach akceptowalne będzie przyjęcie wartości NULL. Każde pole kluczowe jest unikalne, co było niezbędne do osiągnięcia trzeciej postaci normalnej. Wszystkie to sprawia, że spełniony jest warunek zachowania więzów integralności bazy danych.

## Proces denormalizacji – analiza i przykłady

Denormalizacja jest procesem odwrotnym do normalizacji. Powoduje to redundancję danych oraz utratę spójności. Jednak osiągamy szybsze wyszukiwanie danych, co może być szczególnie ważne przy często wykorzystywanych informacjach.

**Możliwe zastosowania denormalizacji:**

* Możliwe jest przeprowadzenie denormalizacji dla encji Adresy. Może to być pomocne w przypadku konieczności częstego i szybkiego wyszukiwania informacji o adresach pracowników, zajezdni, biur itd., jednak spowoduje to bardzo wysoką redundancję, ponieważ dane w takiej formie są wykorzystywane w aż czterech tabelach. Nie zdecydowaliśmy się na tą formę denormalizacji z powodu zbyt dużej ilości powielanych danych,
* Możliwa jest również denormalizacja Bilety i połączenie ich z Pasażerowie, ponieważ Bilety jest połączone jedynie jedną relacją, lecz taki zabieg zdecydowanie obniżyłby czytelność bazy danych oraz kłóciłoby się to z ideą projektowania relacyjnych baz danych, w których każda tabela powinna jednoznacznie identyfikować konkretny projekt.

# Faza fizyczna

## Projekt transakcji i weryfikacja ich wykonalności

|  |  |  |
| --- | --- | --- |
| Transakcja | Weryfikacja wykonalności | Potrzebne dane |
| Podgląd informacji o tramwajach | Wykonalne | Tramwaje |
| Dodawanie/modyfikowanie/usuwanie informacji o tramwajach | Wykonalne | Tramwaje |
| Podgląd informacji o zajezdniach | Wykonalne | Zajezdnie, Adresy |
| Dodawanie/modyfikowanie/usuwanie informacji o zajezdniach | Wykonalne | Zajezdnie, Adresy |
| Podgląd informacji o biurach | Wykonalne | Biura, Adresy |
| Dodawanie/modyfikowanie/usuwanie informacji o biurach | Wykonalne | Biura, Adresy |
| Podgląd informacji o pracownikach | Wykonalne | Pracownicy, Adresy, Biura, Zajezdnie |
| Dodawanie/modyfikowanie/usuwanie informacji o pracownikach | Wykonalne | Pracownicy, Adresy, Biura, Zajezdnie |
| Podgląd informacji o pasażerach | Wykonalne | Pasażerowie, Bilety |
| Podgląd informacji o pracownikach jako kierowcy | Wykonalne | Pracownicy, Kierowcy |
| Dodawanie/modyfikowanie/usuwanie informacji o pracownikach jako kierowcy | Wykonalne | Pracownicy, Kierowcy |
| Podgląd informacji o pracownikach jako o pracownikach obsługi klienta | Wykonalne | Pracownicy, Obslugi, Znajomosci\_jezykow, Jezyki, Poziomy\_znajomosci |
| Dodawanie/modyfikowanie/usuwanie informacji o pracownikach jako o pracownikach obsługi klienta | Wykonalne | Pracownicy, Obslugi, Znajomosci\_jezykow, Jezyki, Poziomy\_znajomosci |
| Podgląd informacji o pracownikach jako o kontrolerach | Wykonalne | Pracownicy, Kontrolerzy |
| Dodawanie/modyfikowanie/usuwanie informacji o pracownikach jako o kontrolerach | Wykonalne | Pracownicy, Kontrolerzy |
| Podgląd informacji o wynagrodzeniu pracowników | Wykonalne | Pracownicy |
| Dodawanie/modyfikowanie/usuwanie informacji o wynagrodzeniu pracowników | Wykonalne | Pracownicy |
| Podgląd informacji o językach obcych | Wykonalne | Znajomosci\_jezykow |
| Dodawanie/modyfikowanie/usuwanie informacji o językach obcych | Wykonalne | Znajomosci\_jezykow |
| Podgląd informacji o znajomości języków obcych przez pracowników obsługi klienta | Wykonalne | Pracownicy, Obslugi, Znajomosci\_jezykow, Jezyki, Poziomy\_znajomosci |
| Dodawanie/modyfikowanie/usuwanie informacji o znajomości języków obcych przez pracowników obsługi klienta | Wykonalne | Pracownicy, Obslugi, Znajomosci\_jezykow, Jezyki, Poziomy\_znajomosci |
| Dodawanie/modyfikowanie/usuwanie informacji o pasażerach | Wykonalne | Tramwaje, Pasazerowie, Bilety |
| Podgląd informacji o biletach | Wykonalne | Bilety |
| Dodawanie/modyfikowanie/usuwanie informacji o biletach | Wykonalne | Bilety |
| Podgląd informacji o reklamach i miejscach reklamowych | Wykonalne | Reklamy |
| Dodawanie/modyfikowanie/usuwanie informacji o reklamach i miejscach reklamowych | Wykonalne | Reklamy |
| Podgląd informacji o tramwajach | Wykonalne | Tramwaje |
| Dodawanie/modyfikowanie/usuwanie informacji o tramwajach | Wykonalne | Tramwaje |

## Strojenie bazy danych – dobór indeksów

W opisywanym projekcie do strojenia bazy danych służyć będą indeksy wygenerowane

automatycznie przy okazji tworzenia relacji między tabelami:

* Wyszukiwanie adresu zajezdni

CREATE INDEX IX\_Adres\_zajezdni ON Zajezdanie (Nr\_adresu)

* Wyszukiwanie pracowników zajezdni

CREATE INDEX IX\_Pracuje\_w\_zajezdni ON Pracownicy (Nr\_zajezdni)

* Wyszukiwanie pracowników biura

CREATE INDEX IX\_Pracuje\_w ON Pracownicy (Nr\_biura)

* Wyszukiwanie adresu pracownika

CREATE INDEX IX\_Adres\_pracownika ON Pracownicy (Nr\_adresu)

* Wyszukiwanie adresu biura

CREATE INDEX IX\_Adres\_biura ON Biura (Nr\_adresu)

* Wyszukiwanie biletów klienta

CREATE INDEX IX\_Posiada ON Bilety (Nr\_klienta)

## Skrypt SQL zakładający bazę danych

/\*

Created: 11/26/2022

Modified: 12/5/2022

Project: Miejskie przedsiebiorstwo tramwajowe

Model: Miejskie przedsiebiorstwo tramwajowe

Company: WEiTI, Politechnika Warszawska

Author: Jan Wojciechowski, Kacper Szaruch

Version: 1.1

Database: Oracle 19c

\*/

-- Create tables section -------------------------------------------------

-- Table Tramwaje

CREATE TABLE Tramwaje(

Nr\_tramwaju Integer NOT NULL,

Czy\_klimatyzacja Char(1 ) NOT NULL,

Czy\_ogrzewanie Char(1 ) NOT NULL,

Czy\_sprawny Char(1 ) NOT NULL,

Data\_przegladu Date NOT NULL,

Pojemnosc Integer NOT NULL,

Ilosc\_miejsc\_siedzacych Integer NOT NULL,

Ilosc\_miejsc\_niepelnosprawnych Integer NOT NULL,

Ilosc\_kasownikow Integer NOT NULL,

Czy\_biletomat Char(1 ) NOT NULL,

Nr\_modelu Integer NOT NULL

)

/

-- Create indexes for table Tramwaje

CREATE INDEX IX\_Relationship2 ON Tramwaje ('Nr\_modelu')

/

-- Add keys for table Tramwaje

ALTER TABLE Tramwaje ADD CONSTRAINT Unique\_Identifier1 PRIMARY KEY (Nr\_tramwaju)

/

-- Table Zajezdanie

CREATE TABLE Zajezdanie(

Nr\_zajezdni Integer NOT NULL,

Czy\_funkcjonuje Char(1 ) NOT NULL,

Pojemnosc Integer NOT NULL,

Godziny\_otwarcia Varchar2(32 ) NOT NULL,

Czy\_myjnia Char(1 ) NOT NULL,

Firma\_ochroniarska Varchar2(40 ),

Ilosc\_tramwajow\_serwisowych Integer NOT NULL,

Czy\_automatyczne\_zwrotnice Char(1 ) NOT NULL,

Czy\_lakiernia Char(1 ) NOT NULL,

Czy\_tokarka\_podtorowa Char(1 ) NOT NULL,

Nr\_adresu Integer NOT NULL

)

/

-- Create indexes for table Zajezdanie

CREATE INDEX IX\_Adres\_zajezdni ON Zajezdanie (Nr\_adresu)

/

-- Add keys for table Zajezdanie

ALTER TABLE Zajezdanie ADD CONSTRAINT Unique\_Identifier2 PRIMARY KEY (Nr\_zajezdni)

/

-- Table Pracownicy

CREATE TABLE Pracownicy(

Nr\_pracownika Integer NOT NULL,

Imie Varchar2(30 ) NOT NULL,

Nazwisko Varchar2(30 ) NOT NULL,

Data\_urodzenia Date NOT NULL,

PESEL Char(11 ) NOT NULL,

Nazwisko\_panienskie\_matki Varchar2(30 ) NOT NULL,

Dowod\_osobisty Varchar2(9 ) NOT NULL,

Czy\_karany Char(1 ) NOT NULL,

Wyksztalcenie Varchar2(30 ) NOT NULL,

Nr\_konta Varchar2(26 ) NOT NULL,

Nr\_telefonu Varchar2(9 ) NOT NULL,

Email Varchar2(40 ),

Wynagrodzenie Number(10,2) NOT NULL,

Plec Varchar2(4 ) NOT NULL

CHECK (Plec IN ('K','M','Inne')),

Nr\_zajezdni Integer NOT NULL,

Nr\_biura Integer NOT NULL,

Nr\_adresu Integer NOT NULL

)

/

-- Create indexes for table Pracownicy

CREATE INDEX IX\_Pracuje\_w\_zajezdni ON Pracownicy (Nr\_zajezdni)

/

CREATE INDEX IX\_Pracuje\_w ON Pracownicy (Nr\_biura)

/

CREATE INDEX IX\_Adres\_pracownika ON Pracownicy (Nr\_adresu)

/

-- Add keys for table Pracownicy

ALTER TABLE Pracownicy ADD CONSTRAINT Unique\_Identifier3 PRIMARY KEY (Nr\_pracownika)

/

-- Table Kierowcy

CREATE TABLE Kierowcy(

Nr\_prawa\_jazdy Varchar2(14 ) NOT NULL,

Data\_waznosci\_prawa\_jazdy Date NOT NULL,

Data\_waznosci\_badan Date NOT NULL,

Nr\_pracownika Integer NOT NULL

)

/

-- Add keys for table Kierowcy

ALTER TABLE Kierowcy ADD CONSTRAINT Unique\_Identifier4 PRIMARY KEY (Nr\_pracownika)

/

-- Table Obslugi

CREATE TABLE Obslugi(

Stanowisko Varchar2(20 ) NOT NULL,

Jezyk Varchar2(60 ) NOT NULL,

Dodatkowe\_kursy Varchar2(400 ),

Nr\_pracownika Integer NOT NULL

)

/

-- Add keys for table Obslugi

ALTER TABLE Obslugi ADD CONSTRAINT Unique\_Identifier5 PRIMARY KEY (Nr\_pracownika)

/

-- Table Kontrolerzy

CREATE TABLE Kontrolerzy(

Licencja Varchar2(30 ) NOT NULL,

Liczba\_wystawionych\_mandatow Integer NOT NULL,

Nr\_pracownika Integer NOT NULL

)

/

-- Add keys for table Kontrolerzy

ALTER TABLE Kontrolerzy ADD CONSTRAINT Unique\_Identifier6 PRIMARY KEY (Nr\_pracownika)

/

-- Table Biura

CREATE TABLE Biura(

Nr\_biura Integer NOT NULL,

Nazwa Varchar2(20 ) NOT NULL,

Czy\_ulatwienia\_dostepu Char(1 ) NOT NULL,

Ilosc\_stanowisk Integer NOT NULL,

Nr\_adresu Integer NOT NULL

)

/

-- Create indexes for table Biura

CREATE INDEX IX\_Adres\_biura ON Biura (Nr\_adresu)

/

-- Add keys for table Biura

ALTER TABLE Biura ADD CONSTRAINT Unique\_Identifier7 PRIMARY KEY (Nr\_biura)

/

-- Table Pasazerowie

CREATE TABLE Pasazerowie(

Nr\_klienta Integer NOT NULL,

Imie Varchar2(20 ),

Nazwisko Varchar2(30 )

)

/

-- Add keys for table Pasazerowie

ALTER TABLE Pasazerowie ADD CONSTRAINT Unique\_Identifier8 PRIMARY KEY (Nr\_klienta)

/

-- Table Bilety

CREATE TABLE Bilety(

Nr\_biletu Integer NOT NULL,

Czy\_ulgowy Char(1 ) NOT NULL,

Data\_zakupu Date,

Data\_skasowania Date,

Wazny\_do Date,

Obslugiwane\_strefy Integer NOT NULL

CHECK (Obslugiwane\_strefy IN ('1','2')),

Czas\_waznosci Date NOT NULL,

Czy\_imienny Char(1 ) NOT NULL,

Nr\_klienta Integer NOT NULL,

Nr\_biura Integer NOT NULL

)

/

-- Create indexes for table Bilety

CREATE INDEX IX\_Posiada ON Bilety (Nr\_klienta)

/

CREATE INDEX IX\_Relationship1 ON Bilety ('Nr\_biura')

/

-- Add keys for table Bilety

ALTER TABLE Bilety ADD CONSTRAINT Unique\_Identifier9 PRIMARY KEY (Nr\_biletu)

/

-- Table Reklamy

CREATE TABLE Reklamy(

Nr\_reklamy Integer NOT NULL,

Format Varchar2(20 ) NOT NULL,

Czy\_podswietlenie Char(1 ) NOT NULL,

Data\_konca Date NOT NULL,

Dlugosc Date,

Lokalizacja Varchar2(50 ) NOT NULL

)

/

-- Add keys for table Reklamy

ALTER TABLE Reklamy ADD CONSTRAINT Unique\_Identifier14 PRIMARY KEY (Nr\_reklamy)

/

-- Table Miejsca\_Postojowe

CREATE TABLE Miejsca\_Postojowe(

Nr\_tramwaju Integer NOT NULL,

Nr\_zajezdni Integer NOT NULL

)

/

-- Table Obslugi\_tramwajow

CREATE TABLE Obslugi\_tramwajow(

Nr\_tramwaju Integer NOT NULL,

Nr\_pracownika Integer NOT NULL

)

/

-- Table Obslugi\_zajezdni

CREATE TABLE Obslugi\_zajezdni(

Nr\_biura Integer NOT NULL,

Nr\_zajezdni Integer NOT NULL

)

/

-- Table Pasazerowie\_tramwajow

CREATE TABLE Pasazerowie\_tramwajow(

Nr\_klienta Integer NOT NULL,

Nr\_tramwaju Integer NOT NULL

)

/

-- Table Gabloty\_reklamowe

CREATE TABLE Gabloty\_reklamowe(

Nr\_tramwaju Integer NOT NULL,

Nr\_reklamy 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(4 ) NOT NULL,

Kod\_pocztowy Char(6 ) NOT NULL

)

/

-- 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 'Nazwa ulicy'

/

COMMENT ON COLUMN Adresy.Nr\_lokalu IS 'Numer lokalu'

/

COMMENT ON COLUMN Adresy.Kod\_pocztowy IS 'Kod pocztowy'

/

-- Table Znajomosci\_jezykow

CREATE TABLE Znajomosci\_jezykow(

Nr\_poziomu Integer NOT NULL,

Nr\_jezyka Integer NOT NULL,

Nr\_pracownika Integer NOT NULL

)

/

-- Create indexes for table Znajomosci\_jezykow

CREATE INDEX IX\_Znane\_jezyki ON Znajomosci\_jezykow (Nr\_poziomu)

/

-- Add keys for table Znajomosci\_jezykow

ALTER TABLE Znajomosci\_jezykow ADD CONSTRAINT PK\_Znajomosci\_jezykow PRIMARY KEY (Nr\_jezyka,Nr\_pracownika)

/

-- Table Jezyki

CREATE TABLE Jezyki(

Nr\_jezyka Integer NOT NULL,

Kod\_jezyka Varchar2(3 ) NOT NULL,

Nazwa Varchar2(20 ) NOT NULL

)

/

-- Add keys for table Jezyki

ALTER TABLE Jezyki ADD CONSTRAINT PK\_Jezyki PRIMARY KEY (Nr\_jezyka)

/

ALTER TABLE Jezyki ADD CONSTRAINT Kod\_jezyka UNIQUE (Kod\_jezyka)

/

-- Table and Columns comments section

COMMENT ON COLUMN Jezyki.Nr\_jezyka IS 'Unikatowy identyfikator języka'

/

COMMENT ON COLUMN Jezyki.Kod\_jezyka IS 'Unikatowy kod jezyka'

/

COMMENT ON COLUMN Jezyki.Nazwa IS 'Nazwa jezyka'

/

-- Table Poziomy\_znajomosci

CREATE TABLE Poziomy\_znajomosci(

Nr\_poziomu Integer NOT NULL,

Kod\_poziomu Varchar2(3 ) NOT NULL,

Opis\_poziomu Varchar2(400 ) NOT NULL

)

/

-- Add keys for table Poziomy\_znajomosci

ALTER TABLE Poziomy\_znajomosci ADD CONSTRAINT PK\_Poziomy\_znajomosci PRIMARY KEY (Nr\_poziomu)

/

ALTER TABLE Poziomy\_znajomosci ADD CONSTRAINT Kod\_poziomu UNIQUE (Kod\_poziomu)

/

-- Table Modele

CREATE TABLE Modele(

Nr\_modelu Integer NOT NULL,

Kod\_modelu Varchar2(15 ) NOT NULL

)

/

-- 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)

/

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

ALTER TABLE Pracownicy ADD CONSTRAINT Pracuje\_w FOREIGN KEY (Nr\_zajezdni) REFERENCES Zajezdanie (Nr\_zajezdni)

/

ALTER TABLE Pracownicy ADD CONSTRAINT Zatrudnia FOREIGN KEY (Nr\_biura) REFERENCES Biura (Nr\_biura)

/

ALTER TABLE Bilety ADD CONSTRAINT Posiada FOREIGN KEY (Nr\_klienta) REFERENCES Pasazerowie (Nr\_klienta)

/

ALTER TABLE Biura ADD CONSTRAINT Biuro\_ma\_adres FOREIGN KEY (Nr\_adresu) REFERENCES Adresy (Nr\_adresu)

/

ALTER TABLE Zajezdanie ADD CONSTRAINT Zajezdnia\_ma\_adres FOREIGN KEY (Nr\_adresu) REFERENCES Adresy (Nr\_adresu)

/

ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik\_ma\_adres FOREIGN KEY (Nr\_adresu) REFERENCES Adresy (Nr\_adresu)

/

ALTER TABLE Znajomosci\_jezykow ADD CONSTRAINT Jezyk\_jest\_znany\_na\_poziomie FOREIGN KEY (Nr\_poziomu) REFERENCES Poziomy\_znajomosci (Nr\_poziomu)

/

ALTER TABLE Znajomosci\_jezykow ADD CONSTRAINT Jezyk\_jest\_znany FOREIGN KEY (Nr\_jezyka) REFERENCES Jezyki (Nr\_jezyka)

/

ALTER TABLE Znajomosci\_jezykow ADD CONSTRAINT Obsluga\_zna\_jezyk FOREIGN KEY (Nr\_pracownika) REFERENCES Obslugi (Nr\_pracownika)

/

ALTER TABLE Bilety ADD CONSTRAINT Sprzedaja FOREIGN KEY (Nr\_biura) REFERENCES Biura (Nr\_biura)

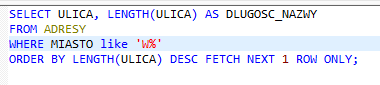
/

ALTER TABLE Tramwaje ADD CONSTRAINT Tramwaj\_jest\_modelu FOREIGN KEY (Nr\_modelu) REFERENCES Modele (Nr\_modelu)

/

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

* Informacja o długości nazwy ulicy o najdłuższej nazwie spośród miast, których nazwa rozpoczyna się na literę „W”



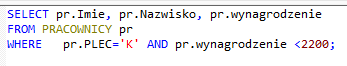


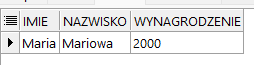
* Informacja o ilości pracowników, których imiona zaczynają się na literę „M”, oraz mają powyżej 20 lat





* Informacja o ilości kobiet, których zarobki są mniejsze niż 2200 zł





* Informacja o imieniu, nazwisku i miejscu zamieszkania pracownika, który był karany





# Bibliografia

W trakcie realizacji projektu pomocne były slajdy wykładowe z przedmiotu Bazy Danych i Big Data (BADA) autorstwa dr hab. inż. Marcina Kowalczyka (Instytut Telekomunikacji na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej).