

BAZY DANYCH I BIG DATA

Dokumentacja

Temat projektu: Baza komunikacji miejskiej

Spis treści

1. Zakres i cel projektu.....	3
2. Definicja systemu.....	3
2.1 Perspektywy użytkowników	4
3. Model konceptualny.....	4
3.1 Definicja zbiorów encji określonych w projekcie (decyzje projektowe).....	4
3.2 Ustalenie związków między encjami i ich typów	5
3.3 Określenie atrybutów i ich dziedzin	5
3.4 Dodatkowe reguły integralnościowe (reguły biznesowe).....	8
3.5 Klucze kandydujące i główne (decyzje projektowe)	8
3.6 Schemat ER na poziomie konceptualnym	8
3.7 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady	8
4. Model logiczny	9
4.1 Charakterystyka modelu relacyjnego.....	9
4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady	10
4.3 Proces normalizacji – analiza i przykłady	10
4.4 Schemat ER na poziomie modelu logicznego.....	11
4.5 Więzy integralności	11
4.6 Proces denormalizacji – analiza i przykłady	11
5. Faza fizyczna	11
5.1 Projekt transakcji i weryfikacja ich wykonalności	11
5.2 Strojenie bazy danych – dobór indeksów	13
5.3 Skrypt SQL zakładający bazę danych.....	14
5.4 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych.....	22

1. Zakres i cel projektu.

Celem projektu jest opracowanie relacyjnej bazy danych na dwóch poziomach: koncepcyjnym i logicznym. Należy także zaimplementować bazę danych w sposób fizyczny. Baza danych zawiera zbiór informacji o bazie komunikacji miejskiej.

Użyte oprogramowanie do stworzenia projektu:

- TOAD Data Modeler 7.3
- SQL Developer 20.4.1
- Lokalna baza danych Oracle Database 19c

Opis założeń funkcjonalnych projektowanej bazy danych:

Biuro Komunikacji Miejskiej zapewnia komunikację miejską w obrębie danego miasta. Oznacza to, że klienci zobowiązują się do kupienia odpowiedniego biletu na przejazd, a kierowca prowadzący pojazd ma za zadanie trzymać się zadanej trasy i odbyć kurs.

W bazie zapisane są wszystkie rodzaje dostępnych do kupienia w danym mieście biletów. Warunkiem korzystania z komunikacji miejskiej jest nabycie biletu.

Baza przechowuje takie dane jak imię, nazwisko, adres klienta. Klienci mogą posiadać ulgę, która pozwala im kupić bilet w niższej cenie. W przypadku kontroli i nieposiadania biletu zostają obciążeni mandatem. W bazie również znajdują się wszystkie dane o wystawionych mandatach – ich data, kwota oraz dane kontrolera.

W bazie jest spis wszystkich pracowników i szczegółowych informacji na ich temat – dane, adres zamieszkania, informacje o wynagrodzeniu oraz posiadane stanowisko – z wyszczególnieniem na kierowcę, którego wyróżnia obowiązkowe prawo jazdy, oraz kontrolera, do którego przypisane jest konkretne urządzenie do sprawdzania biletów.

Kolejną ważną informacją, są dane o wszystkich dostępnych pojazdach oraz ich specyfikacji, a także rozpisane kursy i trasy, które wykonują. Do każdego kursu przypisany jest konkretny pracownik i pojazd, a także data i godzina jego realizacji. Ponadto w bazie zapisane są również wszystkie przystanki i ich dokładne lokalizacje.

2. Definicja systemu

Funkcjonalności:

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o biurach komunikacji miejskiej
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o adresach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o biletach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o kierowcach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o klientach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o kontrolerach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o kursach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o mandatach

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o pocztach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o pojazdach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o pracownikach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o prawach jazdy
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o przystankach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o stanowiskach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o trasach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o ulgach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o wynagrodzeniach
- podgląd/modyfikowanie/dodawanie/usuwanie informacji o kolejności przystanków na danej trasie

2.1 Perspektywy użytkowników

Właściciel – osoba mająca maksymalną kontrolę nad bazą. Jest w stanie modyfikować każdy element w bazie i ma dostęp do wszystkich informacji

Księgowa – może zarządzać danymi związanymi z pracownikami z miasta, w którym jest zatrudniona

Kierowca - może przeglądać informacje odnośnie kursów odbywających się w jego mieście, a także tras, przystanków i pojazdów

Kontroler biletów - ma dostęp do informacji o wystawionych przez siebie mandatach oraz do

Klient – może sprawdzić jakie bilety zakupił, przeglądać przystanki, kursy i ich trasy

3. Model konceptualny

3.1 Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

Baza Komunikacji miejskiej – jest to encja główna, która opisuje daną bazę komunikacji miejskiej

Pracownik – encja przechowująca informacje na temat pracownika BKM

Kontroler biletów - specjalizacja pracownika (dodatkowe informacje o pracowniku, który jest kontrolerem)

Kierowca - specjalizacja pracownika (dodatkowe informacje o pracowniku, który jest kierowcą)

Pojazd – encja przechowująca dane na temat pojazdów np. rok produkcji, rodzaj pojazdu

Kurs – encja przechowująca informacje na temat kursów np. data kursu, id kursu

Przystanek - encja przechowująca informacje o przystanku np. lokalizacja, nazwa

Bilet – encja przechowująca informacje o biletach np. rodzaj biletu, data skasowania

Klient – encja przechowująca informacje o kliencie np. imię, nazwisko

Mandat - encja przechowująca informacje o mandatach np. data wystawienia, wysokość

3.2 Ustalenie związków między encjami i ich typów

Pomiędzy kim relacja	Krotność	Typ uczestnicwa	Opis relacji
BKM - Pracownik	1 - wielu	BKM obowiązkowy Pracownik opcjonalny	Biuro może zatrudniać od 0 do wielu pracowników, a dany pracownik pracuje w jednym biurze.
BKM - Pojazd	1 - wielu	BKM obowiązkowy Pojazd opcjonalny	Biuro może posiadać w swojej flocie od 0 do wielu pojazdów, a dany pojazd przynależy do jednego biura.
BKM - Bilet	1 - wielu	BKM obowiązkowy Bilet opcjonalny	Biuro może obsługiwać od 0 do wielu biletów, a konkretny bilet jest obsługiwany w jednym biurze.
BKM – Kurs	1 – wielu	BKM obowiązkowy Kurs nieobowiązkowy	Biuro obsługuje od 0 do wielu kursów, a dany kurs jest obsługiwany przez jedno biuro
Pojazd - Kurs	1 – wielu	Pojazd obowiązkowy Kurs opcjonalny	Pojazd może brać udział w 0 lub wielu kursach, a dany kurs jest przypisany do konkretnego pojazdu.
Kurs – Przystanek	wiele - wielu	Kurs opcjonalny Przystanek obowiązkowy	W danym kursie występuje od 1 do wielu przystanków, a przez dany przystanek może przejeżdżać od 0 do wielu kursów.
Pojazd – Pracownik	wiele - wielu	Pojazd opcjonalny Pracownik obowiązkowy	Dany pojazd może być obsługiwany przez 1 lub wielu pracowników, a pracownik może obsługiwać 0 lub wiele pojazdów.
Pracownik-Kurs	1 – wielu	Pracownik obowiązkowy Kurs opcjonalny	Pracownik może prowadzić 0 lub wiele kursów, a kurs musi być prowadzony przez jednego pracownika.
Klient – Bilet	1 – wielu	Klient nieobowiązkowy Bilet nieobowiązkowy	Klient może posiadać 0 lub wiele biletów, a bilet może przynależeć do 0 lub 1 klienta.
Klient – Mandat	1 – wielu	Klient obowiązkowy Mandat nieobowiązkowy	Klient może posiadać od 0 do wielu biletów, a dany bilet przynależy do konkretnego klienta.
Kontroler – Mandat	1 – wielu	Kontroler obowiązkowy Mandat nieobowiązkowy	Kontroler może wystawić od 0 do wielu biletów, a dany bilet jest wystawiony przez konkretnego kontrolera

3.3 Określenie atrybutów i ich dziedzin

Biuro komunikacji miejskiej:

Nazwa atrybutu	Typ i dziedzina	Czy obowiązkowy	Opis
ID_biura	SmallInt	Tak	Klucz główny
Nazwa	VarChar(40)	Tak	Nazwa biura
Miasto	VarChar(20)	Tak	Miasto w jakim znajduje się biuro
Data_zalozenia	Date	Tak	Data założenia biura
Strona_WWW	VarChar(50)	Nie	Adres strony internetowej biura

Pracownik:

Nazwa atrybutu	Typ i dziedzina	Czy obowiązkowy	Opis
ID_pracownika	Integer	Tak	Klucz główny
Imie	VarChar(20)	Tak	Imię pracownika
Nazwisko	VarChar(30)	Tak	Nazwisko pracownika
Data_urodzenia	Date	Tak	Data urodzenia pracownika
Zarobki	Money	Tak	Zarobki pracownika
Plec	PlecD	Tak	Płeć pracownika
Data_zatrudnienia	Date	Tak	Data zatrudnienia pracownika
Adres	VarChar(200)	Tak	Adres zamieszkania pracownika
Stanowisko	VarChar(30)	Tak	Stanowisko na jakim pracuje pracownik
Nr_telefonu	VarChar(12)	Tak	Numer telefonu pracownika
Data_zwolnienia	Date	Nie	Data zwolnienia pracownika
PESEL	Character	Nie	Numer PESEL pracownika

Pojazd:

Nazwa atrybutu	Typ i dziedzina	Czy obowiązkowy	Opis
ID_pojazdu	Integer	Tak	Klucz główny
Rodzaj_pojazdu	Rodzaj_pojazduD	Tak	Rodzaj pojazdu
Ilosc_miejsc	Integer	Tak	Ilość dostępnych miejsc
Rodzaj_napedu	Rodzaj_napeduD	Tak	Rodzaj napędu w pojeździe
Rok_produkcji	SmallInt	Tak	Rok produkcji pojazdu

Bilet:

Nazwa atrybutu	Typ i dziedzina	Czy obowiązkowy	Opis
ID_biletu	Integer	Tak	Klucz główny
Rodzaj_biletu	Rodzaj_biletuD	Tak	Rodzaj biletu
Data_skasowania	DateTime	Nie	Data skasowania biletu

Klient:

Nazwa atrybutu	Typ i dziedzina	Czy obowiązkowy	Opis
ID_klienta	Integer	Tak	Klucz główny
Imie	VarChar(20)	Tak	Imie klienta
Nazwisko	VarChar (30)	Tak	Nazwisko klienta
Adres	VarChar(200)	Tak	Adres zamieszkania klienta
Nr_telefonu	Character(12)	Nie	Numer telefonu klienta
Ulga	VarChar(100)	Nie	Ulga przysługująca klientowi

Kurs:

Nazwa atrybutu	Typ i dziedzina	Czy obowiązkowy	Opis
ID_kursu	Integer	Tak	Klucz główny
Czy_nocny	Czy_nocnyD	Tak	Informacja o tym, czy kurs jest nocny
Data	Date	Tak	Data kursu

Przystanek:

Nazwa atrybutu	Typ i dziedzina	Czy obowiązkowy	Opis
ID_przystanku	Integer	Tak	Klucz główny
Nazwa	VarChar(40)	Tak	Nazwa przystanku
Szerokosc_geograficzna	Decimal(8,6)	Tak	Szerokość geograficzna przystanku
Dlugosc_geograficzna	Decimal(8,6)	Tak	Długość geograficzna przystanku
Na_zadanie	Na_zadanieD	Tak	Informacja o tym, czy przystanek jest na żądanie

Mandat:

Nazwa atrybutu	Typ i dziedzina	Czy obowiązkowy	Opis
ID_mandatu	Integer	Tak	Klucz główny
Wysokosc_mandatu	Money	Tak	Wysokość mandatu
Data	Date	Tak	Data wystawienia mandatu

Kontroler Biletów:

Nazwa atrybutu	Typ i dziedzina	Czy obowiązkowy	Opis
Nr_urzadzenia	VarChar(30)	Tak	Numer urządzenia jakim posługuje się kontroler

Kierowca:

Nazwa atrybutu	Typ i dziedzina	Czy obowiązkowy	Opis
Data_waznosci_badan	Date	Tak	Data ważności badań lekarskich
Prawo_jazdy	VarChar(100)	Tak	Informacje odnośnie prawa-jazdy

3.4 Dodatkowe reguły integralnościowe (reguły biznesowe)

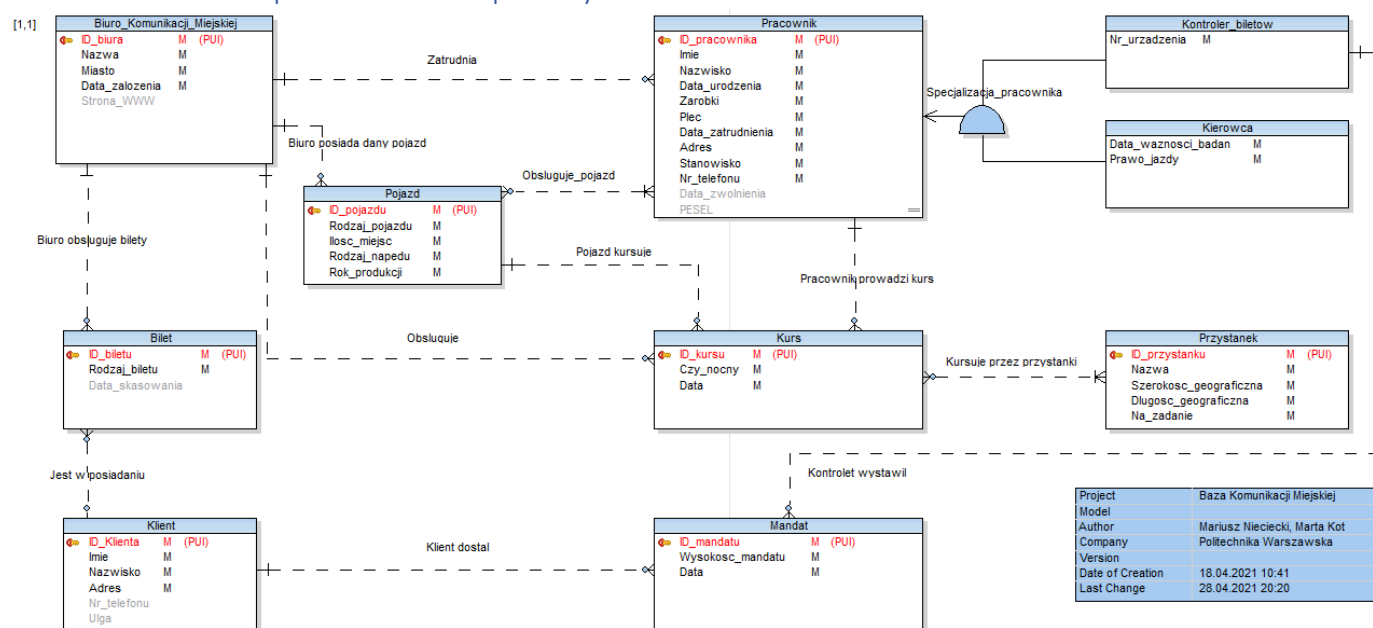
- Klient może mieć wiele biletów, ale nie oznacza to, że będzie mógł z nich korzystać we wszystkich biurach (miastach) – w różnych miastach mogą obowiązywać różne bilety
- Bilet obejmuje tylko jednego klienta
- Kontrola kontrolera w trakcie przejazdu klienta bez ważnego biletu skutkuje wystawieniem mandatu
- Kierowca może prowadzić kursy tylko na terenie miasta, które obsługuje jego biuro

3.5 Klucze kandydujące i główne (decyzje projektowe)

Encja	Klucz główny	Klucz kandydujący
BKM	ID_biura	Nazwa, Miasto
Pracownik	ID_pracownika	PESEL
Przystanek	ID_przystanku	Nazwa, szerokosc_geograficzna+dlugosc_geograficzna
Klient	ID_klienta	Nr_telefonu

W innych encjach jako klucz główny wszędzie przyjęliśmy sztucznie wygenerowane ID, gdyż nie było w nich istotnych kandydatów na klucz główny.

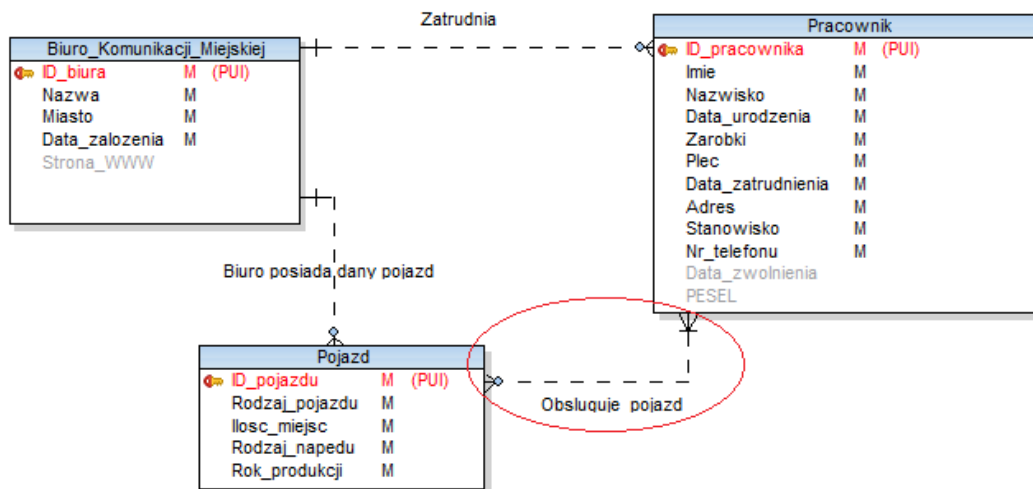
3.6 Schemat ER na poziomie konceptualnym



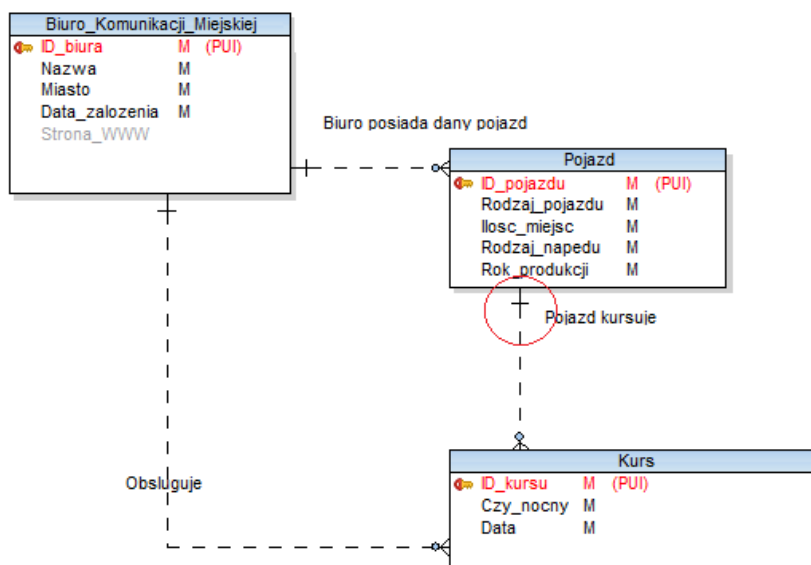
3.7 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

Pułapka wachlarzowa:

W zaprojektowanym przez nas modelu pułapka wachlarzowa mogła pojawić się między encjami Pracownik, Pojazd – BKM. Krotność po stronie BKM dla tych encji wynosi jeden. Aby zabezpieczyć się przed ewentualnymi szkodami wyrządzonymi przez pułapkę wachlarzową (niewiedza, który pracownik obsługuje pojazd) dodano relację między Pracownikiem i Pojazdem – „Obsługuje pojazd”.



W przypadku pułapki szczelinowej – mogłaby ona zaistnieć w naszym projekcie, gdyby istniał kurs, który nie potrzebowałby żadnego pojazdu - doszłoby wtedy do pułapki szczelinowej. Kluczowa byłaby wtedy relacja „Obsługuje” łącząca Kurs i BKM, ponieważ zwalczalaby ten problem.



4. Model logiczny

4.1 Charakterystyka modelu relacyjnego

Model relacyjny stworzyliśmy dzięki przekształceniu modelu konceptualnego za pomocą narzędzi środowiska Toad Data Modeler

Zmiana modelu konceptualnego na logiczny spowodowała że:

- Zostały zmienione typy danych na te, które są obsługiwane przez wybrany przez nas silnik
- W miejscach relacji wiele do wielu pojawiły się tzw. tablice brydżujące (2 relacje jeden do wielu)
- Rozszerzenie specjalizacji pracownik zostało zmienione na encje - otrzymały klucz obcy pochodzący z encji nadrzędnej

4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

Usunięcie związków wiele do wielu:

Kurs-Przystanek: powstanie tablicy brydzącej ze związkami jeden do wielu

Pojazd-Pracownicy: powstanie tablicy brydzącej ze związkami jeden do wielu

4.3 Proces normalizacji – analiza i przykłady

Normalizacja jest to proces, który prowadzi kolejno do schematów, które nie mają powtórzeń, poszczególne wartości zależą od klucza i ostatecznie od niczego innego tylko klucza (brak zależności przechodnich). Taki sposób tworzenia bazy danych zwiększa bezpieczeństwo danych i zmniejsza ryzyko powstania niespójności. Warto zauważyć, że normalizacja nie usuwa danych, tylko zmienia schemat bazy danych.

Przy projektowaniu bazy danych należy zwrócić uwagę, aby spełniała ona 3 postacie normalne:

Pierwsza postać normalna (1PN) – relacja jest w pierwszej postaci normalnej wttw gdy każdy atrybut jest wartością atomową

Druga postać normalna (2PN) – relacja jest w drugiej postaci normalnej wtedy, gdy jest w 1PN i każdy atrybut niekluczowy jest funkcyjnie zależny od całego klucza głównego

Trzecia postać normalna (3PN) – relacja jest w trzeciej postaci normalnej wtedy, gdy jest w 2PN i każdy atrybut niekluczowy jest bezpośrednio zależny funkcyjnie (brak zależności pośrednich) od klucza głównego. Przykładem dążenia do spełniania 3ciej postaci normalnej w naszym projekcie jest dodanie nowych encji w modelu logicznym w miejscu pól wielowartościowych: adresy, poczty, stanowiska, wynagrodzenia, ulgi, trasy.

Adresy → ID_ulgi, miasto, ulica, nr lokalu

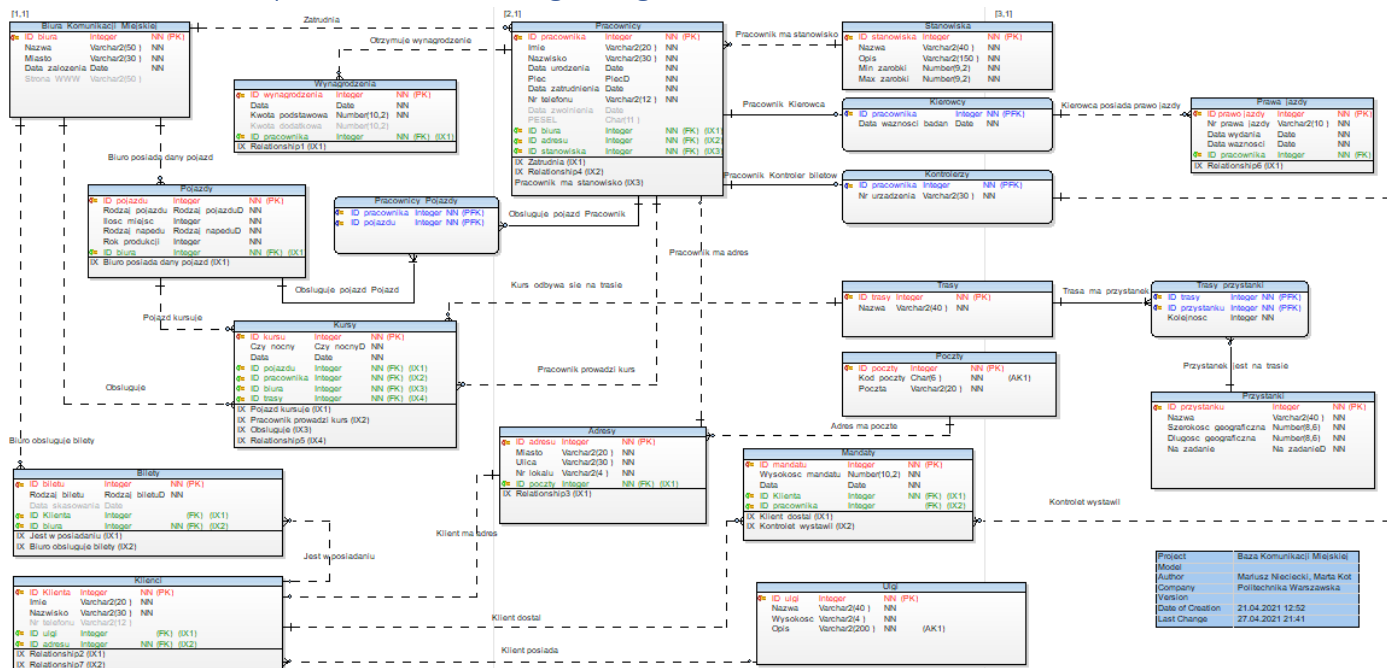
Poczty → ID_poczty, kod poczty, poczta

Stanowiska → ID_stanowiska, nazwa, opis, minimalne zarobki, maksymalne zarobki

Wynagrodzenia → ID_wynagrodzenia, data, kwota podstawowa, kwota dodatkowa

Ulgi → ID_ulgi, nazwa, wysokość, opis

4.4 Schemat ER na poziomie modelu logicznego



4.5 Więzy integralności

Więzy integralności stanowią system reguł zapewniający poprawność związków między rekordami tabel powiązanych relacją i uniemożliwiający przypadkowe usunięcie lub zmianę powiązanych danych. Wynikają z konieczności zapewnienia w bazie spójności danych.

4.6 Proces denormalizacji – analiza i przykłady

Denormalizacja powoduje redundancję danych (powtarzanie się) oraz utratę spójności danych. Plusem stosowania zabiegu redundancji jest to, że zyskujemy na szybkości podczas wyszukiwania danych – jest stosowana aby przyspieszyć wyszukiwanie często wykorzystywanych danych.

W wykonanym przez nas projekcie moglibyśmy dokonać denormalizacji w przypadku encji Klienci – Ułgi oraz Pracownicy – Wynagrodzenia gdybyśmy chcieli zoptymalizować czas wykonywanych przez nas zapytań. Spowodowałoby to jednak redundancję danych (szczególnie w przypadku encji Klienci – Ułgi).

5. Faza fizyczna

5.1 Projekt transakcji i weryfikacja ich wykonalności

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o biurach komunikacji miejskiej **WYKONALNE**

Potrzebne dane z: **Biura_komunikacji_miejskiej**

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o adresach **WYKONALNE**

Potrzebne dane z: **Adresy, Poczty**

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o biletach **WYKONALNE**

Potrzebne dane z: Bilety, Biura_komunikacji_miejskiej, Klienci

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o kierowcach **WYKONALNE**

Potrzebne dane z: Kierowcy, Pracownicy

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o klientach **WYKONALNE**

Potrzebne dane z: Klienci, Ulgi, Adresy

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o kontrolerach **WYKONALNE**

Potrzebne dane z: Kontrolerzy, Pracownicy

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o kursach **WYKONALNE**

Potrzebne dane z: Kursy, Pojazdy, Pracownicy, Biura_komunikacji_miejskiej, Trasy

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o mandatach **WYKONALNE**

Potrzebne dane z: Mandaty, Klienci, Pracownicy

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o pocztach **WYKONALNE**

Potrzebne dane z: Poczty

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o pojazdach **WYKONALNE**

Potrzebne dane z: Pojazdy, Biura_komunikacji_miejskiej

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o pracownikach **WYKONALNE**

Potrzebne dane z: Pracownicy, Biura_komunikacji_miejskiej, Adresy, Stanowiska

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o prawach jazdy **WYKONALNE**

Potrzebne dane z: Prawa_jazdy, Pracownicy

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o przystankach **WYKONALNE**

Potrzebne dane z: Przystanki

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o stanowiskach **WYKONALNE**

Potrzebne dane z: Stanowiska

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o trasach **WYKONALNE**

Potrzebne dane z: Trasy

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o ulgach **WYKONALNE**

Potrzebne dane z: Ulgi

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o wynagrodzeniach **WYKONALNE**

Potrzebne dane z: Wynagrodzenia, Pracownicy

- podgląd/modyfikowanie/dodawanie/usuwanie informacji o kolejności przystanków na danej trasie **WYKONALNE**

Potrzebne dane z: Trasy_przystanki, Trasy, Przystanki

5.2 Strojenie bazy danych – dobór indeksów

- Pracownicy z danego biura

CREATE INDEX IX_Zatrudnia ON Pracownicy (ID_biura)

- Adres pracownika

CREATE INDEX IX_Pracownik_ma_adres ON Pracownicy (ID_adresu)

- Stanowisko pracownika

CREATE INDEX IX_Pracownik_ma_stanowisko ON Pracownicy (ID_stanowiska)

- Bilety klienta

CREATE INDEX IX_Jest_w_posiadaniu ON Bilety (Id_Klienta)

- Z jakiego biura są bilety

CREATE INDEX IX_Biuro_obsługuje_bilety ON Bilety (ID_biura)

- Jaka ulga przysługuje do klienta

CREATE INDEX IX_Klient_posiada ON Klienci (ID_ulgi)

- Adres klienta

CREATE INDEX IX_Klient_mandat ON Klienci (ID_adresu)

- Pojazdy przynależące do biura

CREATE INDEX IX_Biuro_posiada_dany_pojazd ON Pojazdy (ID_biura)

- Pojazd, który bierze udział w kursie

CREATE INDEX IX_Pojazd_kursuje ON Kursy (ID_pojazdu)

- Pracownik, który bierze udział w kursie

CREATE INDEX IX_Pracownik_prowadzi_kurs ON Kursy (ID_pracownika)

- Z jakiego biura jest kurs

CREATE INDEX IX_Obsluguje ON Kursy (ID_biura)

- Przez jaką trasę przejeżdża kurs

CREATE INDEX IX_Kurs_odbywa_sie_na_trasie ON Kursy (Id_trasy)

- Do którego klienta należy mandat

CREATE INDEX IX_Klient_dostal ON Mandaty (Id_Klienta)

- Który kontroler wystawił mandat

CREATE INDEX IX_Kontrolet_wystawil ON Mandaty (ID_pracownika)

- Do jakiej poczty przynależy adres

CREATE INDEX IX_Adres_ma_poczte ON Adresy (ID_poczty)

- Do którego pracownika należy prawo jazdy

CREATE INDEX IX_Kierowca_prawo_jazdy ON Prawa_jazdy (ID_pracownika)

- Do którego pracownika należy wynagrodzenie

CREATE INDEX IX_Otrzymuje_wynagrodzenie ON Wynagrodzenia (ID_pracownika)

5.3 Skrypt SQL zakładający bazę danych

```

/*
Created: 21.04.2021
Modified: 29.04.2021
Project: Baza Komunikacji Miejskiej
Company: Politechnika Warszawska
Author: Mariusz Nieciecki, Marta Kot
Database: Oracle 12c
*/

-- Create tables section -----
-- Table Pracownicy
CREATE TABLE Pracownicy(
  ID_pracownika Integer NOT NULL,
  Imie Varchar2(20 ) NOT NULL,
  Nazwisko Varchar2(30 ) NOT NULL,
  Data_urodzenia Date NOT NULL,
  Plec Char(1 ) NOT NULL
  CONSTRAINT PlecC CHECK (Plec IN ('M','K')),
  Data_zatrudnienia Date NOT NULL,
  Nr_telefonu Varchar2(12 ) NOT NULL,
  Data_zwolnienia Date,
  PESEL Char(11 ),
  ID_biura Integer NOT NULL,
  ID_adresu Integer NOT NULL,
  ID_stanowiska Integer NOT NULL
)
/

-- Create indexes for table Pracownicy
CREATE INDEX IX_Zatrudnia ON Pracownicy (ID_biura)
/
CREATE INDEX IX_Pracownik_ma_adres ON Pracownicy (ID_adresu)
/
CREATE INDEX IX_Pracownik_ma_stanowisko ON Pracownicy (ID_stanowiska)
/

-- Add keys for table Pracownicy
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownicy_PK PRIMARY KEY (ID_pracownika)
/

-- Table and Columns comments section
COMMENT ON COLUMN Pracownicy.ID_pracownika IS 'Unikatowy identyfikator pracownika'

```

```

/
COMMENT ON COLUMN Pracownicy.Imie IS 'Imie pracownika'
/
COMMENT ON COLUMN Pracownicy.Nazwisko IS 'Nazwisko pracownika'
/
COMMENT ON COLUMN Pracownicy.Data_urodzenia IS 'Data urodzenia pracownika'
/
COMMENT ON COLUMN Pracownicy.Plec IS 'Plec pracownika'
/
COMMENT ON COLUMN Pracownicy.Data_zatrudnienia IS 'Data zatrudnienia pracownika'
/
COMMENT ON COLUMN Pracownicy.Nr_telefonu IS 'Numer telefonu pracownika'
/
COMMENT ON COLUMN Pracownicy.Data_zwolnienia IS 'Data zwolnienia pracownika'
/
COMMENT ON COLUMN Pracownicy.PESEL IS 'PESEL pracownika'
/
COMMENT ON COLUMN Pracownicy.ID_biura IS 'Identyfikator biura FK'
/
COMMENT ON COLUMN Pracownicy.ID_adresu IS 'Identyfikator adresu FK'
/
COMMENT ON COLUMN Pracownicy.ID_stanowiska IS 'Identyfikator stanowiska FK'
/
-- Table Biura_Komunikacji_Miejskiej
CREATE TABLE Biura_Komunikacji_Miejskiej(
    ID_biura Integer NOT NULL,
    Nazwa Varchar2(50 ) NOT NULL,
    Miasto Varchar2(30 ) NOT NULL,
    Data_zalozenia Date NOT NULL,
    Strona_WWW Varchar2(50 )
)
/
-- Add keys for table Biura_Komunikacji_Miejskiej
ALTER TABLE Biura_Komunikacji_Miejskiej ADD CONSTRAINT BiuroKomunikacjiMiejskiej_PK PRIMARY KEY (ID_biura)
/
ALTER TABLE Biura_Komunikacji_Miejskiej ADD CONSTRAINT Miasto UNIQUE (Miasto)
/
-- Table and Columns comments section
COMMENT ON COLUMN Biura_Komunikacji_Miejskiej.ID_biura IS 'Unikatowy identyfikator biura'
/
COMMENT ON COLUMN Biura_Komunikacji_Miejskiej.Nazwa IS 'Nazwa biura'
/
COMMENT ON COLUMN Biura_Komunikacji_Miejskiej.Miasto IS 'Miasto w jakim znajduje się biuro'
/
COMMENT ON COLUMN Biura_Komunikacji_Miejskiej.Data_zalozenia IS 'Data zalozenia biura'
/
COMMENT ON COLUMN Biura_Komunikacji_Miejskiej.Strona_WWW IS 'Adres strony internetowej biura'
/
-- Table Bilety

CREATE TABLE Bilety(
    ID_biletu Integer NOT NULL,
    Rodzaj_biletu Varchar2(30 ) NOT NULL
        CONSTRAINT Rodzaj_biletuC CHECK (Rodzaj_biletu IN ('Calodobowy','20 minutowy','Miesieczny','75 minutowy','30 minutowy')),
    Data_skasowania Date,
    Id_Klienta Integer,
    ID_biura Integer NOT NULL
)
/
-- Create indexes for table Bilety
CREATE INDEX IX_Jest_w_posiadaniu ON Bilety (Id_Klienta)
/
CREATE INDEX IX_Biuro_obsluguje_bilety ON Bilety (ID_biura)
/
-- Add keys for table Bilety
ALTER TABLE Bilety ADD CONSTRAINT BiletyPK PRIMARY KEY (ID_biletu)
/
-- Table and Columns comments section
COMMENT ON COLUMN Bilety.ID_biletu IS 'Unikatowy identyfikator biletu'
/
COMMENT ON COLUMN Bilety.Rodzaj_biletu IS 'Rodzaj biletu'
/
COMMENT ON COLUMN Bilety.Data_skasowania IS 'Data skasowania biletu'
/
COMMENT ON COLUMN Bilety.Id_Klienta IS 'Identyfikator klienta FK'
/

```

```

COMMENT ON COLUMN Bilety.ID_biura IS 'Identyfikator biura FK'
/
-- Table Klienci
CREATE TABLE Klienci(
    ID_Klienta Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Nr_telefonu Varchar2(12 ),
    ID_ulgi Integer,
    ID_adresu Integer NOT NULL
)
/
-- Create indexes for table Klienci
CREATE INDEX IX_Klient_posiada ON Klienci (ID_ulgi)
/
CREATE INDEX IX_Klient_mandat ON Klienci (ID_adresu)
/
-- Add keys for table Klienci
ALTER TABLE Klienci ADD CONSTRAINT Klient_PK PRIMARY KEY (ID_Klienta)
/
-- Table and Columns comments section
COMMENT ON COLUMN Klienci.ID_Klienta IS 'Unikatowy identyfikator klienta'
/
COMMENT ON COLUMN Klienci.Imie IS 'Imie klienta'
/
COMMENT ON COLUMN Klienci.Nazwisko IS 'Nazwisko klienta'
/
COMMENT ON COLUMN Klienci.Nr_telefonu IS 'Numer telefonu klienta'
/
COMMENT ON COLUMN Klienci.ID_ulgi IS 'Identyfikator ulgi FK'
/
COMMENT ON COLUMN Klienci.ID_adresu IS 'Identyfikator adresu FK'
/
-- Table Pojazdy
CREATE TABLE Pojazdy(
    ID_pojazdu Integer NOT NULL,
    Rodzaj_pojazdu Varchar2(30 ) NOT NULL
        CONSTRAINT Rodzaj_pojazduC CHECK (Rodzaj_pojazdu IN ('AUTOBUS','TRAMWAJ','METRO')),
    Ilosc_miejsc Integer NOT NULL,
    Rodzaj_napedu Varchar2(20 ) NOT NULL
        CONSTRAINT Rodzaj_napeduC CHECK (Rodzaj_napedu IN ('Silnik spalinowy','Hybrydowy','Elektryczny')),
    Rok_produkcji Integer NOT NULL,
    ID_biura Integer NOT NULL
)
/
-- Create indexes for table Pojazdy
CREATE INDEX IX_Biuro_posiada_dany_pojazd ON Pojazdy (ID_biura)
/
-- Add keys for table Pojazdy

ALTER TABLE Pojazdy ADD CONSTRAINT Pojazdy_PK PRIMARY KEY (ID_pojazdu)
/
-- Table and Columns comments section
COMMENT ON COLUMN Pojazdy.ID_pojazdu IS 'Unikatowy identyfikator pojazdu'
/
COMMENT ON COLUMN Pojazdy.Rodzaj_pojazdu IS 'Rodzaj pojazdu'
/
COMMENT ON COLUMN Pojazdy.Ilosc_miejsc IS 'Ilosc miejsc jaka jest dostepna w danym pojezdzie'
/
COMMENT ON COLUMN Pojazdy.Rodzaj_napedu IS 'Rodzaj napedu w danym pojezdzie'
/
COMMENT ON COLUMN Pojazdy.Rok_produkcji IS 'Rok produkcji pojazdu'
/
COMMENT ON COLUMN Pojazdy.ID_biura IS 'Identyfikator biura FK'
/
-- Table Kursy
CREATE TABLE Kursy(
    ID_kursu Integer NOT NULL,
    Czy_nocny Char(3 ) NOT NULL
        CONSTRAINT Czy_nocnyC CHECK (Czy_nocny IN ('Tak','Nie')),
    Data Date NOT NULL,
    ID_pojazdu Integer NOT NULL,
    ID_pracownika Integer NOT NULL,
    ID_biura Integer NOT NULL,
    ID_trasy Integer
)

```



```

/
-- Create indexes for table Kursy
CREATE INDEX IX_Pojazd_kursuje ON Kursy (ID_pojazdu)
/
CREATE INDEX IX_Pracownik_prowadzi_kurs ON Kursy (ID_pracownika)
/
CREATE INDEX IX_Obsluguje ON Kursy (ID_biura)
/
CREATE INDEX IX_Kurs_odbywa_sie_na_trasie ON Kursy (ID_trasy)
/
-- Add keys for table Kursy
ALTER TABLE Kursy ADD CONSTRAINT KursyPK PRIMARY KEY (ID_kursu)
/
-- Table and Columns comments section
COMMENT ON COLUMN Kursy.ID_kursu IS 'Unikatowy identyfikator kursu'
/
COMMENT ON COLUMN Kursy.Czy_nocny IS 'Czy kurs jest nocny'
/
COMMENT ON COLUMN Kursy.Data IS 'Data kursu'
/
COMMENT ON COLUMN Kursy.ID_pojazdu IS 'Identyfikator pojazdu FK'
/
COMMENT ON COLUMN Kursy.ID_pracownika IS 'Identyfikator pracownika FK'
/
COMMENT ON COLUMN Kursy.ID_biura IS 'Identyfikator biura FK'
/
-- Table Przystanki
CREATE TABLE Przystanki(
    ID_przystanku Integer NOT NULL,
    Nazwa Varchar2(40 ) NOT NULL,
    Szerokosc_geograficzna Number(8,6) NOT NULL,
    Dlugosc_geograficzna Number(8,6) NOT NULL,
    Na_zadanie Char(256 ) NOT NULL
    CONSTRAINT Na_zadanieC CHECK (Na_zadanie IN ('T','N'))
)
/
-- Add keys for table Przystanki
ALTER TABLE Przystanki ADD CONSTRAINT Przystanki_PK PRIMARY KEY (ID_przystanku)
/
-- Table and Columns comments section
COMMENT ON COLUMN Przystanki.ID_przystanku IS 'Unikatowy identyfikator przystanku'
/
COMMENT ON COLUMN Przystanki.Nazwa IS 'Nazwa przystanku'
/
COMMENT ON COLUMN Przystanki.Szerokosc_geograficzna IS 'Szerokosc geograficzna przystanku'
/
COMMENT ON COLUMN Przystanki.Dlugosc_geograficzna IS 'Dlugosc geograficzna przystanku'
/
COMMENT ON COLUMN Przystanki.Na_zadanie IS 'Czy przystanek jest na zadanie'
/
-- Table Kierowcy
CREATE TABLE Kierowcy(
    ID_pracownika Integer NOT NULL,
    Data_waznosci_badan Date NOT NULL
)
/
-- Add keys for table Kierowcy
ALTER TABLE Kierowcy ADD CONSTRAINT Unique_Identifier1 PRIMARY KEY (ID_pracownika)
/
-- Table and Columns comments section
COMMENT ON COLUMN Kierowcy.ID_pracownika IS 'Identyfikator pracowniak PFK'
/
COMMENT ON COLUMN Kierowcy.Data_waznosci_badan IS 'Data ważności badań lekarskich'
/
-- Table Kontrolerzy
CREATE TABLE Kontrolerzy(
    ID_pracownika Integer NOT NULL,
    Nr_urzadzenia Varchar2(30 ) NOT NULL
)
/
-- Add keys for table Kontrolerzy
ALTER TABLE Kontrolerzy ADD CONSTRAINT Unique_Identifier2 PRIMARY KEY (ID_pracownika)
/
-- Table and Columns comments section
COMMENT ON COLUMN Kontrolerzy.ID_pracownika IS 'Identyfikator pracownika PFK'
/

```

```
COMMENT ON COLUMN Kontrolerzy.Nr_urzadzenia IS 'Numer urzadzenia obslugwanego przez pracownika'
```

```
/
```

```
-- Table Mandaty
```

```
CREATE TABLE Mandaty(  
    ID_mandatu Integer NOT NULL,  
    Wysokosc_mandatu Number(10,2) NOT NULL,  
    Data Date NOT NULL,  
    Id_Klienta Integer NOT NULL,  
    ID_pracownika Integer  
)
```

```
/
```

```
-- Create indexes for table Mandaty
```

```
CREATE INDEX IX_Klient_dostal ON Mandaty (Id_Klienta)
```

```
/
```

```
CREATE INDEX IX_Kontrolet_wystawil ON Mandaty (ID_pracownika)
```

```
/
```

```
-- Add keys for table Mandaty
```

```
ALTER TABLE Mandaty ADD CONSTRAINT Mandat_PK PRIMARY KEY (ID_mandatu)
```

```
/
```

```
-- Table and Columns comments section
```

```
COMMENT ON COLUMN Mandaty.ID_mandatu IS 'Unikatowy identyfikator mandatu'
```

```
/
```

```
COMMENT ON COLUMN Mandaty.Wysokosc_mandatu IS 'Wysokosc wystawionego mandatu'
```

```
/
```

```
COMMENT ON COLUMN Mandaty.Data IS 'Data wystawienia mandatu'
```

```
/
```

```
COMMENT ON COLUMN Mandaty.Id_Klienta IS 'Identyfikator klienta FK'
```

```
/
```

```
COMMENT ON COLUMN Mandaty.ID_pracownika IS 'Identyfikator pracownika FK'
```

```
/
```

```
-- Table Trasy
```

```
CREATE TABLE Trasy(  
    ID_trasy Integer NOT NULL,  
    Nazwa Varchar2(40 ) NOT NULL  
)
```

```
/
```

```
-- Table and Columns comments section
```

```
COMMENT ON COLUMN Trasy.ID_trasy IS 'Unikatowy identyfikator trasy'
```

```
/
```

```
COMMENT ON COLUMN Trasy.Nazwa IS 'Nazwa trasy'
```

```
/
```

```
-- Table Pracownicy_Pojazdy
```

```
CREATE TABLE Pracownicy_Pojazdy(  
    ID_pracownika Integer NOT NULL,  
    ID_pojazdu Integer NOT NULL  
)
```

```
/
```

```
-- Table and Columns comments section
```

```
COMMENT ON COLUMN Pracownicy_Pojazdy.ID_pracownika IS 'Identyfikator pracownika PFK'
```

```
/
```

```
COMMENT ON COLUMN Pracownicy_Pojazdy.ID_pojazdu IS 'Identyfikator pojazdu PFK'
```

```
/
```

```
-- Table Ulgi
```

```
CREATE TABLE Ulgi(  
    ID_ulgi Integer NOT NULL,  
    Nazwa Varchar2(40 ) NOT NULL,  
    Wysokosc Varchar2(4 ) NOT NULL,  
    Opis Varchar2(200 ) NOT NULL  
)
```

```
/
```

```
-- Add keys for table Ulgi
```

```
ALTER TABLE Ulgi ADD CONSTRAINT PK_Ulgi PRIMARY KEY (ID_ulgi)
```

```
/
```

```
ALTER TABLE Ulgi ADD CONSTRAINT Opis UNIQUE (Opis)
```

```
/
```

```
-- Table and Columns comments section
```

```
COMMENT ON COLUMN Ulgi.ID_ulgi IS 'Unikatowy identyfikator ulgi'
```

```
/
```

```
COMMENT ON COLUMN Ulgi.Nazwa IS 'Nazwa ulgi'
```

```
/
```

```
COMMENT ON COLUMN Ulgi.Wysokosc IS 'Wysokosc ulgi'
```

```
/
```

```
COMMENT ON COLUMN Ulgi.Opis IS 'Opis do kogo skierowana jest ulga'
```

```
/
```

```
-- Table Adresy
```

```
CREATE TABLE Adresy(  

```

```

ID_adresu Integer NOT NULL,
Miasto Varchar2(20 ) NOT NULL,
Ulica Varchar2(30 ) NOT NULL,
Nr_lokalu Varchar2(4 ) NOT NULL,
ID_poczty Integer NOT NULL
)
/
-- Create indexes for table Adresy
CREATE INDEX IX_Adres_ma_poczte ON Adresy (ID_poczty)
/
-- Add keys for table Adresy
ALTER TABLE Adresy ADD CONSTRAINT PK_Adresy PRIMARY KEY (ID_adresu)
/
-- Table and Columns comments section
COMMENT ON TABLE Adresy IS 'Encja Adresy'
/
COMMENT ON COLUMN Adresy.ID_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'
/
COMMENT ON COLUMN Adresy.ID_poczty IS 'Identyfikator poczty FK'
/
-- Table Poczty
CREATE TABLE Poczty(
ID_poczty Integer NOT NULL,
Kod_poczty Char(6 ) NOT NULL,
Poczta Varchar2(20 ) NOT NULL
)
/
-- Add keys for table Poczty
ALTER TABLE Poczty ADD CONSTRAINT PK_Poczty PRIMARY KEY (ID_poczty)
/
ALTER TABLE Poczty ADD CONSTRAINT Kod_poczty UNIQUE (Kod_poczty)
/
-- Table and Columns comments section
COMMENT ON TABLE Poczty IS 'Relacja slownikowa poczty'
/
COMMENT ON COLUMN Poczty.ID_poczty IS 'Unikatowy identyfikator poczty'
/
COMMENT ON COLUMN Poczty.Kod_poczty IS 'Kod pocztowy'
/
COMMENT ON COLUMN Poczty.Poczta IS 'Lokalizacja poczty'
/
-- Table Stanowiska
CREATE TABLE Stanowiska(
ID_stanowiska Integer NOT NULL,
Nazwa Varchar2(40 ) NOT NULL,
Opis Varchar2(150 ) NOT NULL,
Min_zarobki Number(9,2) NOT NULL,
Max_zarobki Number(9,2) NOT NULL
)
/
-- Add keys for table Stanowiska
ALTER TABLE Stanowiska ADD CONSTRAINT PK_Stanowiska PRIMARY KEY (ID_stanowiska)
/
-- Table and Columns comments section
COMMENT ON TABLE Stanowiska IS 'Relacja slownikowa stanowiska'
/
COMMENT ON COLUMN Stanowiska.ID_stanowiska IS 'Unikatowy indentyfikator stanowiska'
/
COMMENT ON COLUMN Stanowiska.Nazwa IS 'Nazwa stanowiska'
/
COMMENT ON COLUMN Stanowiska.Opis IS 'Opis stanowiska'
/
COMMENT ON COLUMN Stanowiska.Min_zarobki IS 'Minimalne zarobki na stawowisku'
/
COMMENT ON COLUMN Stanowiska.Max_zarobki IS 'Maksymalne zarobki na stanowisku'
/
-- Table Prawa_jazdy
CREATE TABLE Prawa_jazdy(
ID_prawo_jazdy Integer NOT NULL,
Nr_prawa_jazdy Varchar2(10 ) NOT NULL,

```

```

    Data_wydania Date NOT NULL,
    Data_waznosci Date NOT NULL,
    ID_pracownika Integer NOT NULL
)
/
-- Create indexes for table Prawa_jazdy
CREATE INDEX IX_Kierowca_prawo_jazdy ON Prawa_jazdy (ID_pracownika)
/
-- Add keys for table Prawa_jazdy
ALTER TABLE Prawa_jazdy ADD CONSTRAINT PK_Prawa_jazdy PRIMARY KEY (ID_prawo_jazdy)
/
ALTER TABLE Prawa_jazdy ADD CONSTRAINT Nr_prawa_jazdy UNIQUE (Nr_prawa_jazdy)
/
-- Table and Columns comments section
COMMENT ON TABLE Prawa_jazdy IS 'Encja prawa jazdy'
/
COMMENT ON COLUMN Prawa_jazdy.ID_prawo_jazdy IS 'Unikatowy identyfikator prawo jazdy'
/
COMMENT ON COLUMN Prawa_jazdy.Nr_prawa_jazdy IS 'Numer prawa jazdy'
/
COMMENT ON COLUMN Prawa_jazdy.Data_wydania IS 'Data wydania prawo jazdy'
/
COMMENT ON COLUMN Prawa_jazdy.Data_waznosci IS 'Data waznosci prawo jazdy'
/
COMMENT ON COLUMN Prawa_jazdy.ID_pracownika IS 'Identyfikator pracownika FK'
/
-- Table Wynagrodzenia
CREATE TABLE Wynagrodzenia(
    ID_wynagrodzenia Integer NOT NULL,
    Data Date NOT NULL,
    Kwota_podstawowa Number(10,2) NOT NULL,
    Kwota_dodatkowa Number(10,2),
    ID_pracownika Integer NOT NULL
)
/
-- Create indexes for table Wynagrodzenia
CREATE INDEX IX_Otrzymuje_wynagrodzenie ON Wynagrodzenia (ID_pracownika)
/
-- Add keys for table Wynagrodzenia
ALTER TABLE Wynagrodzenia ADD CONSTRAINT PK_Wynagrodzenia PRIMARY KEY (ID_wynagrodzenia)
/
-- Table and Columns comments section
COMMENT ON COLUMN Wynagrodzenia.ID_wynagrodzenia IS 'Unikatowy identyfikator biura'
/
COMMENT ON COLUMN Wynagrodzenia.Data IS 'Data wynagrodzenia'
/
COMMENT ON COLUMN Wynagrodzenia.Kwota_podstawowa IS 'Kwota podstawowa wynagrodzenia'
/
COMMENT ON COLUMN Wynagrodzenia.Kwota_dodatkowa IS 'Kwota dodatkowa'
/
COMMENT ON COLUMN Wynagrodzenia.ID_pracownika IS 'Identyfikator pracownika FK'
/
-- Table Trasy_przystanki
CREATE TABLE Trasy_przystanki(
    ID_trasy Integer NOT NULL,
    ID_przystanku Integer NOT NULL,
    Kolejnosc Integer NOT NULL
)
/
-- Add keys for table Trasy_przystanki
ALTER TABLE Trasy_przystanki ADD CONSTRAINT PK_Trasy_przystanki PRIMARY KEY (ID_przystanku,ID_trasy)
/
-- Table and Columns comments section
COMMENT ON COLUMN Trasy_przystanki.ID_trasy IS 'Identyfikator trasy'
/
COMMENT ON COLUMN Trasy_przystanki.ID_przystanku IS 'Identyfikator przystanku PFK'
/
COMMENT ON COLUMN Trasy_przystanki.Kolejnosc IS 'Kolejnosc przystankow na trasie '
/
-- Create foreign keys (relationships) section -----
ALTER TABLE Pracownicy ADD FOREIGN KEY (ID_biura) REFERENCES Biura_Komunikacji_Miejskiej (ID_biura)
/
ALTER TABLE Bilety ADD FOREIGN KEY (Id_Klienta) REFERENCES Klienci (ID_Klienta)
/
ALTER TABLE Bilety ADD FOREIGN KEY (ID_biura) REFERENCES Biura_Komunikacji_Miejskiej (ID_biura)
/

```

```
ALTER TABLE Pojazdy ADD FOREIGN KEY (ID_biura) REFERENCES Biura_Komunikacji_Miejskiej (ID_biura)
/
ALTER TABLE Kursy ADD FOREIGN KEY (ID_pojazdu) REFERENCES Pojazdy (ID_pojazdu)
/
ALTER TABLE Kursy ADD FOREIGN KEY (ID_pracownika) REFERENCES Pracownicy (ID_pracownika)
/
ALTER TABLE Mandaty ADD FOREIGN KEY (Id_Klienta) REFERENCES Klienci (ID_Klienta)
/
ALTER TABLE Mandaty ADD FOREIGN KEY (ID_pracownika) REFERENCES Kontrolerzy (ID_pracownika)
/
ALTER TABLE Kursy ADD FOREIGN KEY (ID_biura) REFERENCES Biura_Komunikacji_Miejskiej (ID_biura)
/
ALTER TABLE Klienci ADD FOREIGN KEY (ID_ulgi) REFERENCES Ulgi (ID_ulgi)
/
ALTER TABLE Adresy ADD FOREIGN KEY (ID_poczty) REFERENCES Poczty (ID_poczty)
/
ALTER TABLE Pracownicy ADD FOREIGN KEY (ID_adresu) REFERENCES Adresy (ID_adresu)
/
ALTER TABLE Pracownicy ADD FOREIGN KEY (ID_stanowiska) REFERENCES Stanowiska (ID_stanowiska)
/
ALTER TABLE Prawa_jazdy ADD FOREIGN KEY (ID_pracownika) REFERENCES Kierowcy (ID_pracownika)
/
ALTER TABLE Klienci ADD FOREIGN KEY (ID_adresu) REFERENCES Adresy (ID_adresu)
/
ALTER TABLE Wynagrodzenia ADD FOREIGN KEY (ID_pracownika) REFERENCES Pracownicy (ID_pracownika)
/
ALTER TABLE Kursy ADD FOREIGN KEY (ID_trasy) REFERENCES Trasy (ID_trasy)
/
ALTER TABLE Trasy_przystanki ADD FOREIGN KEY (ID_trasy) REFERENCES Trasy (ID_trasy)
/
ALTER TABLE Trasy_przystanki ADD FOREIGN KEY (ID_przystanku) REFERENCES Przystanki (ID_przystanku)
```

5.4 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

- Wyświetl nazwę ulgi i ilość klientów jaka ją posiada.

```
SELECT count(id_klienta), u.nazwa
FROM klienci k JOIN ulgi u on(k.id_ulgi=u.id_ulgi)
GROUP BY u.nazwa;
```

	ILOSC_KLIENTOW	NAZWA
1	6	Uczen
2	3	Dziecko
3	2	Rencista
4	2	Emeryt
5	5	Student
6	2	Niepelnosprawny

- Wyświetl imię, nazwisko i pesel pracowników, którzy są Konserwatorami

```
SELECT p.imie, p.nazwisko, p.pesel, s.nazwa nazwa_stanowiska
FROM pracownicy p JOIN stanowiska s ON(s.id_stanowiska=p.id_stanowiska)
WHERE s.nazwa='Konserwator';
```

	IMIE	NAZWISKO	PESEL	NAZWA_STANOWISKA
1	Alex	Wolanski	95082633333	Konserwator
2	Wiktoria	Lutynska	95121287938	Konserwator
3	Adam	Kamin	90030423456	Konserwator

- Wyświetl wszystkie kobiety zatrudnione w biurach, które urodziły się po 1990r.

```
SELECT * FROM PRACOWNICY
WHERE plec like 'K' AND extract(year from data_urodzenia)>1990;
```

ID_PRACOWNIKA	IMIE	NAZWISKO	DATA_URODZENIA	PLEC	DATA_ZATRUDNIENIA	NR_TELEFONU	DATA_ZWOLNIENIA	PESEL	ID_BIURA	ID_ADRESU	ID_STANOWISKA
1	112	Wiktoria	Lutynska	95/12/12	K	12/12/12	+48555213549 (null)	95121287938	103	30	101
2	113	Angelika	Michalowska	94/03/22	K	08/04/12	+48754111111 21/02/20	94032269523	104	39	107
3	114	Oliwia	Bak	92/02/02	K	12/12/12	+48792354813 (null)	92020287412	104	40	107
4	123	Henryka	Siebiatynska	95/08/26	K	10/04/04	+48993615777 (null)	95082686492	103	27	104
5	130	Klaudia	Filipczuk	98/12/27	K	10/04/04	+48681384913 (null)	98122788000	104	26	106
6	131	Wieslawa	Zamojda	95/12/12	K	10/04/04	+48538483118 (null)	95121221212	104	44	107
7	140	Karolina	Wnorowska	95/08/26	K	10/04/04	+48963525252 (null)	95082633333	103	30	105

- Wyświetl imię, nazwisko sumę pieniężną mandatów jakie dostał oraz ilość mandatów.

```
SELECT k.imie,k.nazwisko, sum(m.wysokosc_mandatu), count(m.id_mandatu) ilosc_mandatow
FROM klienci k join mandaty m ON(k.id_klienta=m.id_klienta)
GROUP BY k.imie,k.nazwisko;
```

	IMIE	NAZWISKO	SUM(M.WYSOKOSC_MANDATU)	ILOSC_MANDATOW
1	Izydora	Poplawska	200	1
2	Julia	Krawat	350	2
3	Adam	Abacki	450	2
4	Dominik	Dabacki	200	1
5	Daria	Nowak	400	2
6	Kamil	Chrost	180	1
7	Edyta	Kwasniewski	150	1
8	Ireneusz	Grzmot	200	2
9	Kornelia	Frankowska	260	2
10	Eugeniusz	Kowal	140	1

- Wyświetl skład floty (z podziałem na rodzaje pojazdów) jakimi dysponują biura. Pogrupuj wynik według nazwy biura

```
SELECT b.nazwa, count(p.id_pojazdu) ilosc_pojazdow, p.rodzaj_pojazdu
FROM biura_komunikacji_miejskiej b JOIN pojazdy p ON (b.id_biura=p.id_biura)
GROUP BY b.nazwa, p.rodzaj_pojazdu
Order by b.nazwa;
```

	NAZWA	ILOSC_POJAZDOW	RODZAJ_POJAZDU
1	Bialostocki przewoźnik	6	AUTOBUS
2	Warszawski Transport Publiczny	6	METRO
3	Warszawski Transport Publiczny	7	TRAMWAJ
4	Warszawski Transport Publiczny	6	AUTOBUS

- Wyświetl ilość pracowników z konkretnych miejscowości, w których mieszka więcej niż 1 pracownik.

```
SELECT count(p.id_pracownika), a.miasto
FROM pracownicy p JOIN adresy a ON(p.id_adresu=a.id_adresu)
GROUP BY a.miasto
HAVING count(p.id_pracownika)>1;
```

	COUNT(P.ID_PRACOWNIKA)	MIASTO
1	17	Warszawa
2	2	Radzymin
3	2	Augustow
4	13	Bialystok