BAZY DANYCH I BIG DATA

PROJEKT nr 1

Sprawozdanie

**Temat:** Spółdzielnia mieszkaniowa

**Autorzy:**

Maciej Kaczkowski, 300660

Paweł Berentowicz, 300481

Spis treści

[**1.** **Zakres i cel projektu** 3](#_Toc89210285)

[**2.** **Definicja systemu** 4](#_Toc89210286)

[**3.** **Model konceptualny** 6](#_Toc89210287)

[3.1. Definicja zbiorów encji określonych w projekcie oraz określenie atrybutów i ich dziedzin 6](#_Toc89210288)

[3.2. Ustalenie związków między encjami oraz ich typów 9](#_Toc89210289)

[3.3. Klucze kandydujące i główne 10](#_Toc89210290)

[3.4. Schemat ER na poziomie konceptualnym 11](#_Toc89210291)

[3.5. Pułapki szczelinowe i wachlarzowe 11](#_Toc89210292)

[**4.** **Model logiczny** 12](#_Toc89210293)

[4.1. Usunięcie właściwości niekompatybilnych z modelem relacyjnym 12](#_Toc89210294)

[4.2. Proces normalizacji 13](#_Toc89210295)

[4.2. 1. SpółdzielnieMieszkaniowe: 13](#_Toc89210296)

[4.2. 2. StronyInternetowe: 13](#_Toc89210297)

[4.2.3. Adresy: 13](#_Toc89210298)

[4.2. 4. Zarzady: 13](#_Toc89210299)

[4.2.5. CzlonkowieZarzadu: 14](#_Toc89210300)

[4.2.6. Pracownicy: 14](#_Toc89210301)

[4.2.7. Stanowiska: 14](#_Toc89210302)

[4.2.8. Wynagrodzenia: 14](#_Toc89210303)

[4.2.9. Umowy: 14](#_Toc89210304)

[4.2.10. Specjalizacje pracowników: 14](#_Toc89210305)

[4.2.11. Klienci: 15](#_Toc89210306)

[4.2.12. DziałkiBudowlane: 15](#_Toc89210307)

[4.2.13. DomyJednorodzinne: 15](#_Toc89210308)

[4.2.14. Bloki: 15](#_Toc89210309)

[4.2.15. Mieszkania: 15](#_Toc89210310)

[4.3. Schemat ER na poziomie modelu logicznego 15](#_Toc89210311)

[4.4. Więzy integralności 15](#_Toc89210312)

[4.5. Proces denormalizacji 16](#_Toc89210313)

[**5.** **Model fizyczny** 17](#_Toc89210314)

[5.1. Projekt transakcji i weryfikacja ich wykonalności 17](#_Toc89210315)

[5.2. Strojenie bazy danych poprzez dobór indeksów 17](#_Toc89210316)

[5.2.1. Klienci 17](#_Toc89210317)

[5.2.2. Pracownicy 17](#_Toc89210318)

[5.2.3. Mieszkania 17](#_Toc89210319)

[5.3. Skrypt SQL generujący bazę danych 18](#_Toc89210320)

[5.4. Przykłady zapytań i poleceń SQL odnoszących się do bazy dany 18](#_Toc89210321)

[5.4.1. Odczytanie informacji o klientach mieszkających w danym bloku 18](#_Toc89210322)

[5.4.1. Odczytanie imion i nazwisk viceprezesów w aktualnym zarządzie 18](#_Toc89210323)

[5.4.2. Odczytanie dat podpisania umowy oraz dat urodzenia wśród aktualnie zatrudnionych pracowników 18](#_Toc89210324)

[**6.** **Załączniki** 19](#_Toc89210325)

[6.1. Załącznik 1: zal\_1\_model\_konceptualny.png 19](#_Toc89210326)

[6.2. Załącznik 2: zal\_2\_model\_logiczny.png 21](#_Toc89210327)

[6.3. Załącznik 3: DLL\_script.SQL 23](#_Toc89210328)

[**7.** **Bibliografia** 47](#_Toc89210329)

# **Zakres i cel projektu**

Celem pierwszej części projektu jest zaprojektowanie oraz zaimplementowanie relacyjnej bazy danych na poziomach konceptualnym i logicznym oraz jej fizyczna implementacja. Baza danych ma za zadanie umożliwić obsługę i zarządzanie spółdzielnią mieszkaniową. Zostanie ona oparta o system zarządzania bazami danych firmy Oracle.

**1.1. Wykorzystane oprogramowanie**

* Oracle 19c DBMS
* Oracle SQL Developer
* TOAD Data Modeler 7.2
* Git 2.31.0.windows.1

**1.2. Ogólny opis**

Realizowany projekt dotyczy spółdzielni mieszkaniowej. Spółdzielnia ta zajmuje się zarządzaniem danego osiedla oraz zaspokajaniem potrzeb mieszkaniowych członków i ich rodzin. Realizuje ona takie zadania jak nabywanie nowych budynków w celu ustanowienia, na rzecz członków spółdzielczych, lokatorskich lub własnościowych praw do znajdujących się w tych budynkach lokali mieszkalnych, budowanie lub nabywanie domów, lub nabywanie budynków w celu wynajmowania lub sprzedaży znajdujących się w tych budynkach lokali mieszkalnych lub lokali o innym przeznaczeniu. Zarządzanie (wynajem, remonty) już posiadanymi budynkami mieszkalnymi, nabywaniem nowych. W tym celu spółdzielnia prowadzi bazę dotyczącą mieszkań wchodzących w skład poszczególnych budynków, która jest również podstawą przy dokonywaniu wyceny mieszkania do zakupu/wynajmu.

# **Definicja systemu**

**2.1. Przykładowe funkcjonalności systemu**

* tworzenie, modyfikacja, podgląd, usuwanie (CRUD) danych personalnych pracowników
* CRUD danych personalnych klientów
* CRUD obiektów zarządzanych przez spółdzielnię - domów, bloków, działek
* możliwość rozszerzenia struktury bazy o dodatkowe informacje (np. jeśli spółdzielnia zacznie zajmować się wynajmem bud dla psów zostanie utworzona relacja “BudaDlaPsa”)

**2.2. Perspektywy użytkowników**

2.2.1. Administrator

Administrator ma dostęp do wszystkich funkcji systemu. Posiada uprawnienia administratora bazy danych Oracle, a zatem może modyfikować jej strukturę (np. dodawanie/ usuwanie nowych relacji), dodawać, usuwać, modyfikować i odczytywać wszystkie dane.

2.2.2. Księgowa/y

Księgowa/y ma dostęp do danych o pracownikach, o ile dotyczą one finansów. Zatem takie dane jak data podpisania umowy, data rozwiązania umowy, wynagrodzenia są dla niej/niego dostępne, a dane takie jak PESEL - nie. Oprócz tego ma dostęp do wszelkich danych związanych z kosztami i zyskami związanymi z zarządzanymi nieruchomościami oraz danymi o sobie.

2.2.3. Członek zarządu

Ma dostęp do wszystkich danych o pracownikach oraz nieruchomościach (odczytywanie, modyfikacja), ale nie może modyfikować struktury bazy danych.

2.2.4. Administrator nieruchomości

Ma dostęp do danych o sobie (modyfikacja, odczytywanie), a także do danych nieruchomości, którymi zarządza. W tym przypadku ma dostęp do wszystkich operacji CRUD - może dodawać, odczytywać, modyfikować i usuwać dane.

2.2.5. Pracownik Sprzątający

Ma dostęp do danych o sobie (modyfikacja, odczytywanie), a także do danych dotyczących bloków (ale nie jego mieszkańców), będących w posiadaniu Spółdzielni (odczytywanie).

2.2.6. Inny pracownik

Ma dostęp do danych o sobie - może je modyfikować oraz odczytywać.

2.2.7. Klient

Może odczytywać dane o sobie, a także je modyfikować. Ma dostęp do danych mieszkania, tak długo jak je zamieszkuje.

# **Model konceptualny**

## 3.1. Definicja zbiorów encji określonych w projekcie oraz określenie atrybutów i ich dziedzin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Spółdzielnia Mieszkaniowa – główna encja** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Numer\_spoldzielni\_PK | SmallInt | T | T | T | Numer jednoznacznie identyfikujący spółdzielnię |
| Adres\_spoldzielni | AdresD | T | T | N | Adres spółdzielni |
| REGON | Character(10) | T | T | T | Nr REGON spółdzielni |
| KRS | Character(9) | T | T | T | Nr KRS spółdzielni |
| NIP | Character(10) | T | T | T | Nr NIP spółdzielni |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Zarząd – reprezentuje encję zarządzającą Spółdzielnią** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Numer\_zarzadu\_PK | SmallInt | T | T | T | Numer jednoznacznie identyfikujący dany zarząd |
| Skarbnik | SmallInt | T | T | N | Reprezentuje skarbnika zarządu |
| Prezes | SmallInt | T | T | N | Reprezentuje Prezesa zarządu |
| Vice\_prezes | SmallInt | T | T | N | Reprezentuje Vice-Prezesa zarządu |
| Rozpoczęcie\_kadencji | Date | T | N | T | Data rozpoczęcia kadencji zarządu |
| Planowane\_zakonczenie\_kandencji | Date | T | N | T | Planowana data zakończenia kadencji zarządu |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pracownik – reprezentuje encję pracownika spółdzielni** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Numer\_pracownika\_PK | Integer | T | T | T | Numer jednoznacznie identyfikujący danego pracownika |
| Pensja\_miesieczna | Money | T | N | N | Reprezentuje wysokość pensji pracownika |
| Umowa\_podpisana | Date | T | N | T | Data podpisania umowy przez pracownika |
| Umowa\_rozwiazana | Date | N | N | T | Data rozwiązania umowy przez pracownika |
| plec | PlecD | T | N | T | Płeć pracownika |
| PESEL | Character(11) | N | T | T | Nr PESEL pracownika |
| Numer\_telefonu | VarChar(9) | N | T | T | Nr telefonu pracownika |
| Stanowisko | VarChar(512) | T | N | N | Stanowisko zajmowane przez pracownika |
| Adres\_pracownika | AdresD | T | T | N | Adres pracownika |
| Data\_urodzenia | Date | N | N | T | Data urodzenia pracownika |
| Email\_pracownika | VarChar(512) | N | T | T | Adres e-mail pracownika |
| Imie | ImieD | T | N | T | Imię pracownika |
| Nazwisko | NazwiskoD | T | N | T | Nazwisko pracownika |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Klient – reprezentuje encję klienta spółdzielni** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Numer\_klienta\_PK | Integer | T | T | T | Numer jednoznacznie identyfikujący danego klienta |
| Imie | ImieD | T | N | T | Imię danego klienta |
| Nazwisko | NazwiskoD | T | N | T | Nazwisko danego klienta |
| Numer\_telefonu | VarChar(12) | N | N | T | Numer telefonu danego klienta |
| Adres\_klienta | AdresD | T | T | N | Adres danego klienta |
| PESEL | Character(11) | N | T | T | Nr PESEL klienta |
| Data\_urodzenia | Date | N | N | T | Data urodzenia Klienta |
| Od\_kiedy\_klient | Date | N | N | T | Data, kiedy klient stał się klientem spółdzielni |
| Płec | PlecD | T | N | T | Płeć klienta |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Blok – reprezentuje encję bloku należącego do spółdzielni** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Numer\_bloku\_PK | Integer | T | T | T | Numer jednoznacznie identyfikujący dany blok |
| Ilosc\_mieszkan | Integer | T | N | T | Ilość mieszkań znajdujących się w danym bloku |
| Adres\_bloku | AdresD | T | T | N | Adres bloku |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mieszkanie – reprezentuje encję mieszkania zawierającego się w bloku** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Numer\_mieszkania\_PK | Integer | T | T | T | Numer jednoznacznie identyfikujący dane mieszkanie |
| Ilosc\_pokoi | SmallInt | T | N | T | Ilość pokoi znajdujących się w danym mieszkaniu |
| Ilosc\_lazienek | SmallInt | T | N | T | Ilość łazienek znajdujących się w danym mieszkaniu |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dom Jednorodzinny – reprezentuje encję domu jednorodzinnego należącego do spółdzielni** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Numer\_domu\_PK | Integer | T | T | T | Numer jednoznacznie identyfikujący dany dom |
| Powierzchnia | Float(2) | T | N | T | Powierzchnia domu |
| Adres\_domu | AdresD | T | T | N | Adres domu |
| Ilosc\_pieter | SmallInt | T | N | T | Ilość pięter domu |
| Ilosc\_pokoi | SmallInt | N | N | T | Ilość pokoi w domu |
| Ilosc\_lazienek | SmallInt | N | N | T | Ilość łazienek w domu |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Działka Budowlana – reprezentuje encję działki budowlanej należącej do spółdzielni** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Numer\_dzialki\_PK | Integer | T | T | T | Numer jednoznacznie identyfikujący działkę |
| Powierzchnia | Float(10) | N | N | T | Powierzchnia działki |
| Adres\_dzialki | AdresD | T | T | N | Adres działki |
| Media | MediaD | N | N | T | Media (woda, gaz, prąd) |
| Garaz | Boolean | N | N | T | Określa, czy działka ma garaż |
| Ksiega\_wieczysta | Boolean | T | T | T | Określa, czy księga wieczysta działki jest aktualna |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Strona Internetowa – reprezentuje encję strony internetowej należącej do spółdzielni** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Numer\_strony\_PK | SmallInt | T | T | T | Numer jednoznacznie identyfikujący stronę internetową |
| Admnistrator | VarChar(256) | T | N | N | Opisuje administratora strony internetowej |
| Adres\_url | VarChar(256) | T | T | T | Adres URL strony internetowej |

Specjalizacje Pracowników:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pracownik Księgowości – reprezentuje encję pracownika księgowości** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Dostep\_do\_danych\_niejawnych | VarChar(512) | T | N | N | Opisuje do jakich danych niejawnych ma dostęp dany pracownik księgowości |
| Numer\_uprawnien | VarChar(64) | T | T | T | Numer uprawnień |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pracownik Administracji– reprezentuje encję pracownika administracji** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Zarzadzane\_bloki | VarChar(512) | N | N | T | Opisuje zakres zarządzania blokami |
| Zarzadzane\_domy | VarChar(512) | N | N | T | Opisuje zakres zarządzania domami |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pracownik Sprzątający– reprezentuje encję pracownika administracji** | | | | | |
| Nazwa atrybutu | Typ i dziedzina | Czy obowiązkowy | Czy unikatowy | Czy prosty | Opis |
| Zarzadzane\_bloki | VarChar(512) | T | N | T | Opisuje sprzątane bloki |

## 3.2. Ustalenie związków między encjami oraz ich typów

Ogólnie rzecz biorąc, w modelu bazy danych występują prawie wszystkie rodzaje związków pomiędzy encjami. Przeważające liczebnie są związki typu 1:n (jeden do wielu. Co charakterystyczne dla modelu konceptualnego - występują związki typu n:m (wiele do wielu). Są one niekompatybilne z modelem relacyjnym, jednak ich wystąpienie było spodziewane, ponieważ często oddają one związki występujące w świecie rzeczywistym. Problem niekompatybilności z modelem relacyjnym zostanie zaadresowany w dalszej części projektu. Wszystkie związki mają krotność 2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parent Entity** | **Child Entity** | **Cardinality** |
| Zatrudnia | SpoldzielniaMieszkaniowa | Pracownik | 1...1 - 0...m |
| Jest\_zarzadzana\_przez | SpoldzielniaMieszkaniowa | Zarzad | 1...1 - 1...1 |
| Posiada\_strone | SpoldzielniaMieszkaniowa | StronaInternetowa | 1...1 - 0...m |
| Posiada\_dzialke | SpoldzielniaMieszkaniowa | DzialkaBudowlana | 1...1 - 0...m |
| Posiada\_dom | SpoldzielniaMieszkaniowa | DomJednorodzinny | 1...1 - 0...m |
| Posiada\_blok | SpoldzielniaMieszkaniowa | Blok | 1...1 - 0...m |
| Zawiera\_mieszkania | Blok | Mieszkanie | 1...1 - 1...m |
| Zarzadza\_domem | AdministratorNieruchomosci | Dom | 1...1 - 0...m |
| Zarzadza\_blokiem | AdministratorNieruchomosci | Blok | 1...1 - 0...m |
| Zamieszkuje\_dom | Klient | Dom | 0...1 - 0...1 |
| Zamieszkuje\_mieszkanie | Klient | Blok | 0...1 - 0...1 |
| Posiada\_skarbnika | PracownikKsiegowosci | Zarzad | 1...1 – 1...1 |
| Posiada\_umowe | Klient | SpoldzielniaMieszkaniowa | 0...n – 1...1 |
| Sprzata\_blok | PracownikSprzatajacy | Blok | 1...n – 0...m |
| Zawiera\_mieszkania | Blok | Mieszkanie | 1...1 – 1...m |

## 3.3. Klucze kandydujące i główne

Zdecydowaliśmy się na użycie kluczy sztucznych, w celu poprawy czytelności i spójności oraz uniknięcia nietypowych błędów, które mogą się pojawić w przypadku użycia kluczy naturalnych (np. pomyłka przy wprowadzaniu numeru PESEL spowoduje złamanie zasady unikatowości kluczy). Innymi kluczami kandydującymi były numery takie jak: REGON, NIP, KRS (w przypadku spółdzielni) oraz PESEL (w przypadku ludzi). Warto jednak zauważyć, że w większości przypadków encji nie ma dobrych naturalnych kluczy kandydujących, zatem i tak jest konieczne użycie kluczy sztucznych.

|  |  |
| --- | --- |
| **Entity** | **Primary Key (PK)** |
| SpoldzielniaMieszkaniowa | numer\_spoldzielni\_PK |
| Zarzad | numer\_zarzadu\_PK |
| Klient | numer\_klienta\_PK |
| Pracownik | numer\_pracownika\_PK |
| PracownikKsiegowosci | numer\_pracownika\_PK |
| Sprzedawca | numer\_pracownika\_PK |
| AdministratorNieruchomosci | numer\_pracownika\_PK |
| DzialkaBudowlana | numer\_dzialki\_PK |
| DomJednorodzinny | numer\_domu\_PK |
| Blok | numer\_bloku\_PK |
| Mieszkanie | numer\_mieszkania\_PK |
| StronaInternetowa | numer\_strony\_PK |
| Zarzad | numer\_zarzadu\_PK |

## 3.4. Schemat ER na poziomie konceptualnym

Zobacz: zal\_1\_model\_konceptualny.png

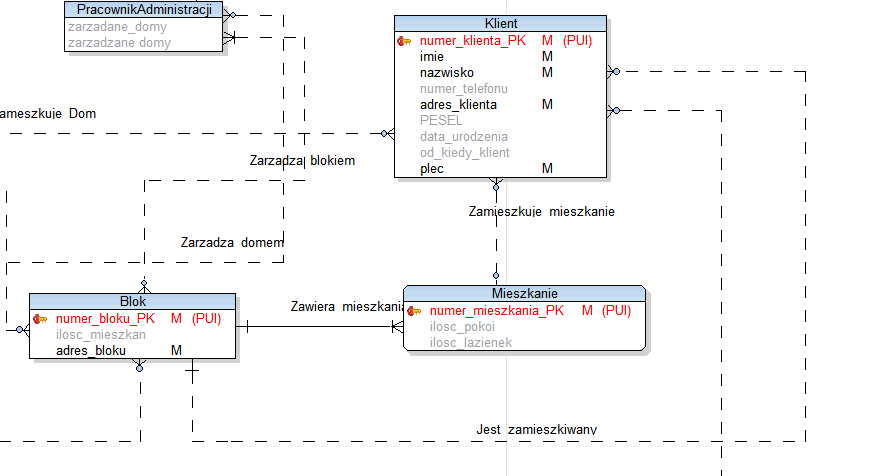
## 3.5. Pułapki szczelinowe i wachlarzowe

3.5.1. Wachlarzowa

W zaproponowanym modelu konceptualnym nie stwierdziliśmy nigdzie występowania problemu pułapki wachlarzowej.

3.5.2. Szczelinowa

W zaproponowanym modelu konceptualnym zauważyliśmy, że sugerowane jest istnienie związku pomiędzy zbiorami encji “Blok” i “Klient”, jednak nie istnieją żadne ścieżki łączące wystąpienia tych encji, czyli mamy do czynienia z pułapką szczelinową. Problem ten rozwiązano dodając związek pomiędzy wspomnianymi wyżej relacjami:



# **Model logiczny**

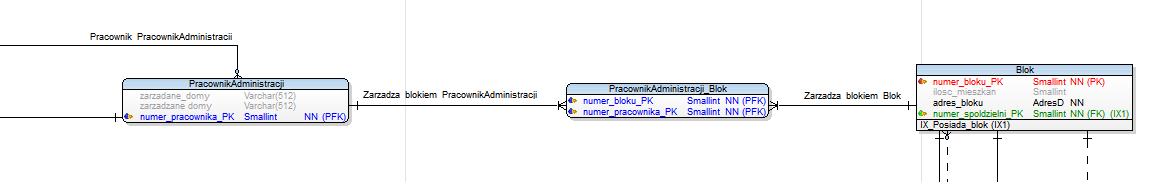
## 4.1. Usunięcie właściwości niekompatybilnych z modelem relacyjnym

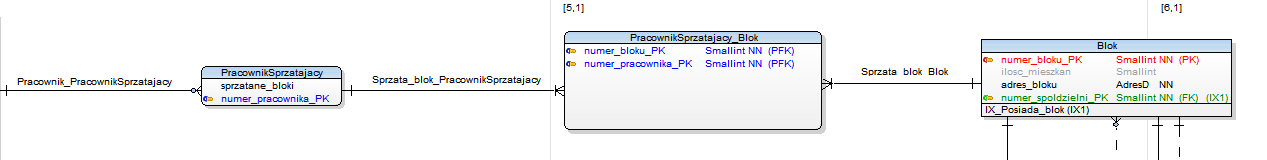
Aby przejść od zaproponowanego modelu konceptualnego do logicznego modelu relacyjnego podjęto próbę usunięcia niekompatybilności modelu konceptualnego z modelem relacyjnym poprzez następujące działania:

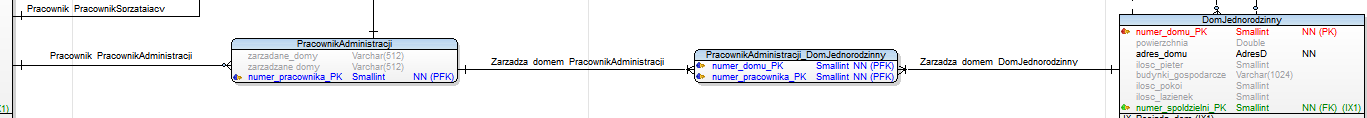
* związki wiele do wielu zastąpiono tzw. tablicami “bridge’ującymi”
* nazwa każdej encji została zmieniona na liczbę mnogą w celu odróżnienia relacji od encji
* identyfikujące atrybuty stały się kluczami głównymi tabeli, pozostałe atrybuty stały się niegłównymi atrybutami tabeli

W modelu konceptualnym znajdowały się 3 związki m:n (wielu do wielu):

* Sprzata\_blok
* Zarzadza\_domem
* Zaradza\_blokiem

W modelu relacyjnym zostały one zastąpione poniższymi tablicami bridge’ującymi:  






# 

## 4.2. Proces normalizacji

Poniżej przedstawiono zmiany, których dokonano na typ etapie względem poszczególnych relacji z modelu konceptualnego, celem usunięcia właściwości niekompatybilnych z modelem relacyjnym. Opisano m .in. zmianę atrybutów będących polami segmentowymi jako osobne relacje, modyfikację/zmianę konkretnych atrybutów itp.

### 4.2. 1. SpółdzielnieMieszkaniowe:

Jako, że atrybut ‘Adres’ jest polem segmentowym, postanowiono wyodrębnić go jako osobną relację, mając również na uwadze, że taki sam atrybut pojawia się również w innych relacjach, dzięki czemu zapewniona będzie większa wygoda i skalowalność. Po utworzeniu nowej, wspomnianej relacji usunięto pierwotny atrybut.

### 4.2. 2. StronyInternetowe:

Pole “Administrator” jest zarówno polem segmentowym (imie, nazwisko, data rozpoczęcia administrowania, itp.) jak i wielowartościowym (więcej niż jeden administrator) - konieczne jest stworzenie osobnej relacji “Administratorzy”, gdzie zostaną wyniesione ich dane oraz linkowane z “macierzystą” relacją. Na koniec usunięto atrybut ‘Administratorzy’ z pierwotnej relacji.

### 4.2.3. Adresy:

Wszystkie atrybuty są atomowe, natomiast mamy do czynienia z powtarzającą się grupą, a mianowicie “Kod poczty” oraz “Poczta” (jeśli dane obiekty/ludzie są z tego samego regionu). W kontekście 1PN można się tego jeszcze nie doszukać, jednak w przypadku 3PN już tak - ‘Nr\_adresu’ wskazuje na ‘Kod\_poczty’, a ‘Kod\_poczty’ wskazuje na Poczta. Dlatego powinna się tutaj pojawić dodatkowa relacja, jaką jest relacja słownikowa - “Poczty”.

### 4.2. 4. Zarzady:

W tym przypadku widzimy, że pola “Skrabnik”, “Prezes” i “Vice prezes” są polami segmentowymi (oraz wielowartościowymi), podobnie jak wcześniej w przypadku relacji ‘StronyInternetowe’, dlatego analogicznie zostanie utworzona tutaj nowa i linkowana relacja “CzłonekwieZarzadu”.

### 4.2.5. CzlonkowieZarzadu:

Utworzone pole “Rola” wymusza albo stworzenie nowej dziedziny albo stworzenie nowej relacji słownikowej, na co zdecydowano się w tym przypadku. Utworzona relację słownikową “Rola” i usunięto następnie atrybut “Rola” w pierwotnej relacji.

### 4.2.6. Pracownicy:

W przypadku atrybutu “Stanowisko” występuje identyczna sytuacja jak powyżej, dlatego utworzoną nową relację słownikową (co sprzyja dynamicznemu przydzielaniu stanowisk, nie mamy na “sztywno” zadeklarowanej dziedziny), o takiej samej nazwie jak wspomniany atrybut, i dodano linkowanie do pierwotnej relacji.  
Pole “pensja\_miesieczna” również jest pole segmentowanym, dlatego utworzono relację “Wynagrodzenie” z odpowiednimi atrybutami i dziedzinami, które linkowano następnie do tej relacji. Zamiast atrybutów “umowa podpisana” oraz “umowa rozwiązana” postanowiono stworzyć osobną relację “Umowa” - z myślą, że tak relacja będzie wykorzystana również w przypadku innych związków.

### 4.2.7. Stanowiska:

Relacja w takiej postaci jak pokazano spełnia już warunki 3PN.

### 4.2.8. Wynagrodzenia:

Relacja w takiej postaci jak pokazano spełnia już warunki 3PN.

### 4.2.9. Umowy:

Relacja w takiej postaci jak pokazano spełnia już warunki 3PN.

### 4.2.10. Specjalizacje pracowników:

***PracownicyKsięgowści:***

Atrybut “dostep\_do\_danych\_niejawnych” (finansowych) jest polem segmentowym - w tym celu stworzono osobną relację “DaneNiejawne”i linkowano ją.

***PracownicyAdministracji:***

Relacja w takiej postaci jak pokazano spełnia już warunki 3PN.

***PracownicySprzątajacy:***

Relacja w takiej postaci jak pokazano spełnia już warunki 3PN.

### 4.2.11. Klienci:

Usunięto atrybut “adres\_klienta” i dodano linkowanie do utworzonej wcześniej, oddzielnej relacji “Adresy”. Usunięto atrybut “od\_kiedy\_klient”, zamiast tego linkowano stworzoną na tym etapie relację “Umowy”.

### 4.2.12. DziałkiBudowlane:

Relacja w takiej postaci jak pokazano spełnia już warunki 3PN.

### 4.2.13. DomyJednorodzinne:

Relacja w takiej postaci jak pokazano spełnia już warunki 3PN.

### 4.2.14. Bloki:

Relacja w takiej postaci jak pokazano spełnia już warunki 3PN.

### 4.2.15. Mieszkania:

Relacja w takiej postaci jak pokazano spełnia już warunki 3PN.

# 4.3. Schemat ER na poziomie modelu logicznego

Zobacz: zal\_2\_model\_logiczny.png

# 4.4. Więzy integralności

Integralność zapewniono za pomocą użycia oznaczeń UNIQUE oraz NOT NULL. Klucze główne i obce, są UNIQUE NOT NULL. Inne atrybuty, których brak mógłby mieć wpływ na działanie bazy danych lub nie ma sensu logicznego (np. działka bez adresu, klient bez nazwiska) zostały oznaczone jako NOT NULL.

# 4.5. Proces denormalizacji

W modelu nie wykonywano denormalizacji – ze względu na brak faktycznych wymagań odnoszących się do obciążenia (np. ilość zapytań/ godzinę), a także rodzajów zapytań baza danych została utrzymana w 3PN, w celu zachowania elastyczności i możliwości rozbudowy o dalsze relacje.

# **Model fizyczny**

Przed utworzeniem modelu fizycznego wykonano sprawdzenie modelu za pomocą opcji „Verify Model” oraz utworzono sekwencje *autoincrement* dla każdej głównej relacji.

## 5.1. Projekt transakcji i weryfikacja ich wykonalności

|  |  |  |
| --- | --- | --- |
| **Transakcja** | **Potrzebne zasoby** | **Czy wykonalna** |
| Dodawanie, modyfikacja, podgląd, usuwanie (CRUD) danych klientów | Klienci, Bloki, Umowy, Mieszkania, Adresy | Tak |
| CRUD danych pracowników | Pracownicy, Wynagrodzenia, Stanowiska, DaneNiejawne, Blok | Tak |
| CRUD danych o posiadanych przez spółdzielnię obiektach | Bloki, Domy, Mieszkania, PracownicyAdministracji, PracownicySprzątający, Adresy, Poczty | Tak |
| CRUD danych o spółdzielni i jej zarządzie | SpółdzielnieMieszkaniowe, Zarządy, CzłonkowieZarządu, Role | Tak |

## 5.2. Strojenie bazy danych poprzez dobór indeksów

### 5.2.1. Klienci

CREATE INDEX IX\_zamieszkuje\_blok ON Klienci (numer\_bloku\_PK);

CREATE INDEX IX\_zamieszkuje\_dom ON Klienci (numer\_dom\_PK);

CREATE INDEX IX\_jest\_klientem\_spoldzielni ON Klienci (numer\_spoldzielni\_PK);

### 5.2.2. Pracownicy

CREATE INDEX IX\_pracuje\_na\_stanowisku ON Pracownicy (numer\_stanowiska);

CREATE INDEX IX\_jest\_pracownikiem\_spoldzielni ON Pracownicy (numer\_spoldzielni\_PK);

### 5.2.3. Mieszkania

CREATE INDEX IX\_jest\_w\_bloku ON Mieszkania (numer\_bloku\_PK);

## 5.3. Skrypt SQL generujący bazę danych

Zobacz: zal\_3\_skrypt\_generacyjny.sql

## 5.4. Przykłady zapytań i poleceń SQL odnoszących się do bazy dany

### 5.4.1. Odczytanie informacji o klientach mieszkających w danym bloku

SELECT \* FROM Klienci WHERE Nr\_bloku\_PK=19;

### 5.4.1. Odczytanie imion i nazwisk viceprezesów w aktualnym zarządzie

SELECT imie, nazwisko FROM CzlonkowieZarzadu

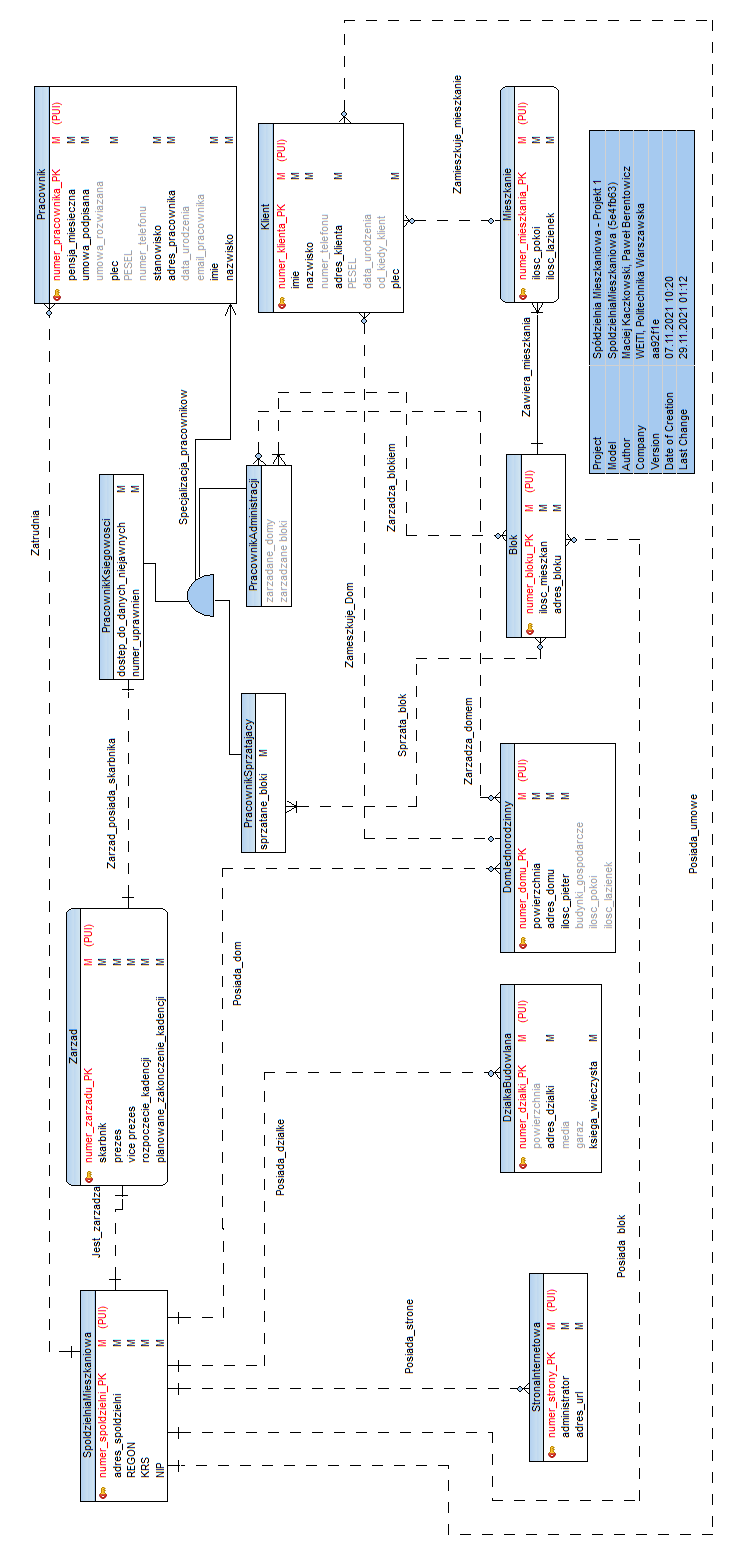
JOIN Role ON CzlonkowieZarzadu.Nr\_roli = Role.Nr\_roli WHERE Role.Nazwa IS ‘viceprezes’;

### 5.4.2. Odczytanie dat podpisania umowy oraz dat urodzenia wśród aktualnie zatrudnionych pracowników

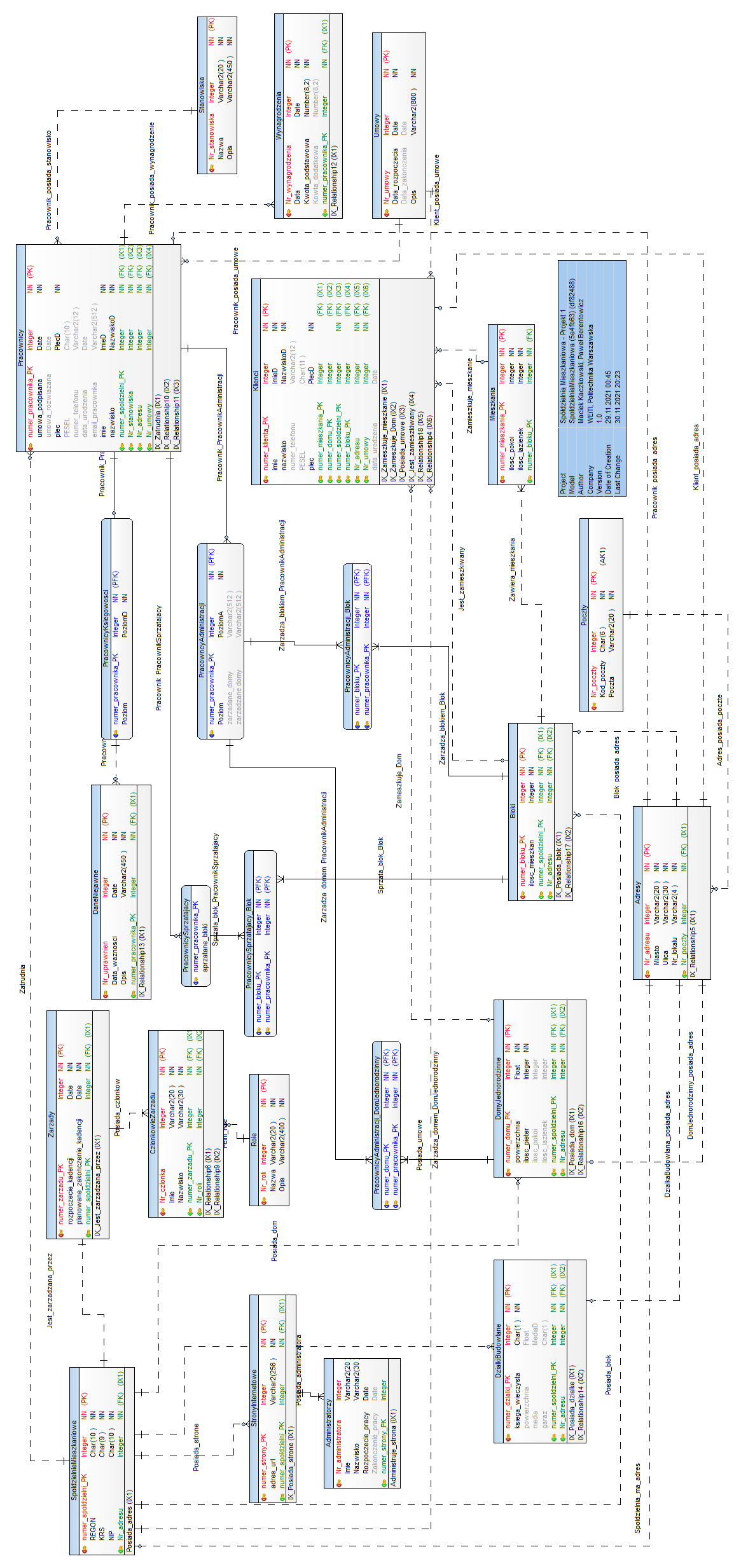
SELECT umowa\_podpisana, data\_urodzenia FROM Pracownicy WHERE umowa\_rozwiazana IS NULL;

# **Załączniki**

# 6.1. Załącznik 1: zal\_1\_model\_konceptualny.png

****

## 6.2. Załącznik 2: zal\_2\_model\_logiczny.png

****

# 

## 6.3. Załącznik 3: zal\_3\_skrypt\_generacyjny.sql

/\*

Created: 2021-11-29

Modified: 2021-11-29

Model: SpoldzielniaMieszkaniowa (5e4fb63) (df82488)

Database: Oracle 19c

\*/

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

CREATE SEQUENCE "NrSpoldzielniSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrZarzaduSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrUprawnienSeq"

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrRoliSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrAdministratoraSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrStronySeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrCzlonkaSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrDzialkiSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrDomuSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrBlokuSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrAdresuSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrPocztySeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrMieszkaniaSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrKlientaSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE "NrPracownikaSeq"

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

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

-- Table SpoldzielnieMieszkaniowe

CREATE TABLE "SpoldzielnieMieszkaniowe"(

"numer\_spoldzielni\_PK" Integer NOT NULL,

"REGON" Char(10 ) NOT NULL,

"KRS" Char(9 ) NOT NULL,

"NIP" Char(10 ) NOT NULL,

"Nr\_adresu" Integer NOT NULL

)

/

-- Create indexes for table SpoldzielnieMieszkaniowe

CREATE INDEX "IX\_Relationship4" ON "SpoldzielnieMieszkaniowe" ("Nr\_adresu")

/

-- Add keys for table SpoldzielnieMieszkaniowe

ALTER TABLE "SpoldzielnieMieszkaniowe" ADD CONSTRAINT "numer\_spoldzielni\_PK" PRIMARY KEY ("numer\_spoldzielni\_PK")

/

-- Table Zarzady

CREATE TABLE "Zarzady"(

"numer\_zarzadu\_PK" Integer NOT NULL,

"rozpoczecie\_kadencji" Date NOT NULL,

"planowane\_zakonczenie\_kadencji" Date NOT NULL,

"numer\_spoldzielni\_PK" Integer NOT NULL

)

/

-- Create indexes for table Zarzady

CREATE INDEX "IX\_Jest\_zarzadzana\_przez" ON "Zarzady" ("numer\_spoldzielni\_PK")

/

-- Add keys for table Zarzady

ALTER TABLE "Zarzady" ADD CONSTRAINT "numer\_zarzadu\_PK" PRIMARY KEY ("numer\_zarzadu\_PK")

/

-- Table Bloki

CREATE TABLE "Bloki"(

"numer\_bloku\_PK" Integer NOT NULL,

"ilosc\_mieszkan" Integer NOT NULL,

"numer\_spoldzielni\_PK" Integer NOT NULL,

"Nr\_adresu" Integer NOT NULL

)

/

-- Create indexes for table Bloki

CREATE INDEX "IX\_Posiada\_blok" ON "Bloki" ("numer\_spoldzielni\_PK")

/

CREATE INDEX "IX\_Relationship17" ON "Bloki" ("Nr\_adresu")

/

-- Add keys for table Bloki

ALTER TABLE "Bloki" ADD CONSTRAINT "numer\_bloku\_PK" PRIMARY KEY ("numer\_bloku\_PK")

/

-- Table Mieszkania

CREATE TABLE "Mieszkania"(

"numer\_mieszkania\_PK" Integer NOT NULL,

"ilosc\_pokoi" Integer NOT NULL,

"ilosc\_lazienek" Integer NOT NULL,

"numer\_bloku\_PK" Integer NOT NULL

)

/

-- Add keys for table Mieszkania

ALTER TABLE "Mieszkania" ADD CONSTRAINT "numer\_mieszkani\_PK" PRIMARY KEY ("numer\_mieszkania\_PK")

/

-- Table DzialkiBudowlane

CREATE TABLE "DzialkiBudowlane"(

"numer\_dzialki\_PK" Integer NOT NULL,

"ksiega\_wieczysta" Char(1 ) NOT NULL,

"powierzchnia" Float,

"media" Varchar2(256 ),

"garaz" Char(1 ),

"numer\_spoldzielni\_PK" Integer NOT NULL,

"Nr\_adresu" Integer NOT NULL

)

/

-- Create indexes for table DzialkiBudowlane

CREATE INDEX "IX\_Posiada\_dzialke" ON "DzialkiBudowlane" ("numer\_spoldzielni\_PK")

/

CREATE INDEX "IX\_Relationship14" ON "DzialkiBudowlane" ("Nr\_adresu")

/

-- Add keys for table DzialkiBudowlane

ALTER TABLE "DzialkiBudowlane" ADD CONSTRAINT "numer\_dzialki\_PK" PRIMARY KEY ("numer\_dzialki\_PK")

/

-- Table Pracownicy

CREATE TABLE "Pracownicy"(

"numer\_pracownika\_PK" Integer NOT NULL,

"umowa\_podpisana" Date NOT NULL,

"umowa\_rozwiazana" Date,

"plec" Char(1 ) NOT NULL,

"PESEL" Char(10 ),

"numer\_telefonu" Varchar2(12 ),

"data\_urodzenia" Date,

"email\_pracownika" Varchar2(512 ),

"imie" Varchar2(64 ) NOT NULL,

"nazwisko" Varchar2(64 ) NOT NULL,

"numer\_spoldzielni\_PK" Integer NOT NULL,

"Nr\_stanowiska" Integer NOT NULL,

"Nr\_adresu" Integer NOT NULL,

"Nr\_umowy" Integer NOT NULL

)

/

-- Create indexes for table Pracownicy

CREATE INDEX "IX\_Zatrudnia" ON "Pracownicy" ("numer\_spoldzielni\_PK")

/

CREATE INDEX "IX\_Relationship10" ON "Pracownicy" ("Nr\_stanowiska")

/

CREATE INDEX "IX\_Relationship11" ON "Pracownicy" ("Nr\_adresu")

/

CREATE INDEX "IX\_Relationship3" ON "Pracownicy" ("Nr\_umowy")

/

-- Add keys for table Pracownicy

ALTER TABLE "Pracownicy" ADD CONSTRAINT "numer\_pracownika\_PK" PRIMARY KEY ("numer\_pracownika\_PK")

/

-- Table Klienci

CREATE TABLE "Klienci"(

"numer\_klienta\_PK" Integer NOT NULL,

"imie" Varchar2(64 ) NOT NULL,

"nazwisko" Varchar2(64 ) NOT NULL,

"numer\_telefonu" Varchar2(12 ),

"PESEL" Char(11 ),

"plec" Char(1 ) NOT NULL,

"numer\_mieszkania\_PK" Integer,

"numer\_domu\_PK" Integer,

"numer\_spoldzielni\_PK" Integer NOT NULL,

"numer\_bloku\_PK" Integer,

"Nr\_adresu" Integer NOT NULL,

"Nr\_umowy" Integer NOT NULL,

"data\_urodzenia" Date

)

/

-- Create indexes for table Klienci

CREATE INDEX "IX\_Zamieszkuje\_mieszkanie" ON "Klienci" ("numer\_mieszkania\_PK")

/

CREATE INDEX "IX\_Zameszkuje\_Dom" ON "Klienci" ("numer\_domu\_PK")

/

CREATE INDEX "IX\_Posiada\_umowe" ON "Klienci" ("numer\_spoldzielni\_PK")

/

CREATE INDEX "IX\_Jest\_zamieszkiwany" ON "Klienci" ("numer\_bloku\_PK")

/

CREATE INDEX "IX\_Relationship18" ON "Klienci" ("Nr\_adresu")

/

CREATE INDEX "IX\_Relationship4" ON "Klienci" ("Nr\_umowy")

/

-- Add keys for table Klienci

ALTER TABLE "Klienci" ADD CONSTRAINT "numer\_klienta\_PK" PRIMARY KEY ("numer\_klienta\_PK")

/

-- Table StronyInternetowe

CREATE TABLE "StronyInternetowe"(

"numer\_strony\_PK" Integer NOT NULL,

"adres\_url" Varchar2(256 ) NOT NULL,

"numer\_spoldzielni\_PK" Integer NOT NULL

)

/

-- Create indexes for table StronyInternetowe

CREATE INDEX "IX\_Posiada\_strone" ON "StronyInternetowe" ("numer\_spoldzielni\_PK")

/

-- Add keys for table StronyInternetowe

ALTER TABLE "StronyInternetowe" ADD CONSTRAINT "numer\_strony\_PK" PRIMARY KEY ("numer\_strony\_PK")

/

-- Table DomyJednorodzinne

CREATE TABLE "DomyJednorodzinne"(

"numer\_domu\_PK" Integer NOT NULL,

"powierzchnia" Float NOT NULL,

"ilosc\_pieter" Integer NOT NULL,

"ilosc\_pokoi" Integer,

"ilosc\_lazienek" Integer,

"numer\_spoldzielni\_PK" Integer NOT NULL,

"Nr\_adresu" Integer NOT NULL

)

/

-- Create indexes for table DomyJednorodzinne

CREATE INDEX "IX\_Posiada\_dom" ON "DomyJednorodzinne" ("numer\_spoldzielni\_PK")

/

CREATE INDEX "IX\_Relationship16" ON "DomyJednorodzinne" ("Nr\_adresu")

/

-- Add keys for table DomyJednorodzinne

ALTER TABLE "DomyJednorodzinne" ADD CONSTRAINT "numer\_domu\_PK" PRIMARY KEY ("numer\_domu\_PK")

/

-- Table PracownicySprzatajacy

CREATE TABLE "PracownicySprzatajacy"(

"numer\_pracownika\_PK" Integer NOT NULL,

"sprzatane\_bloki" Varchar2(512 ) NOT NULL

)

/

-- Add keys for table PracownicySprzatajacy

ALTER TABLE "PracownicySprzatajacy" ADD CONSTRAINT "Unique\_Identifier5" PRIMARY KEY ("numer\_pracownika\_PK")

/

-- Table PracownicyKsiegowosci

CREATE TABLE "PracownicyKsiegowosci"(

"numer\_pracownika\_PK" Integer NOT NULL,

"Poziom" Varchar2(30 ) NOT NULL

)

/

-- Add keys for table PracownicyKsiegowosci

ALTER TABLE "PracownicyKsiegowosci" ADD CONSTRAINT "Unique\_Identifier6" PRIMARY KEY ("numer\_pracownika\_PK")

/

-- Table and Columns comments section

COMMENT ON COLUMN "PracownicyKsiegowosci"."Poziom" IS 'Określa stopień Księgowego (Młodszy/Zwykły/Starszy/Główny Księgowy)'

/

-- Table PracowncyAdministracji

CREATE TABLE "PracowncyAdministracji"(

"numer\_pracownika\_PK" Integer NOT NULL,

"Poziom" Varchar2(30 ) NOT NULL,

"zarzadane\_domy" Varchar2(512 ),

"zarzadzane domy" Varchar2(512 )

)

/

-- Add keys for table PracowncyAdministracji

ALTER TABLE "PracowncyAdministracji" ADD CONSTRAINT "Unique\_Identifier7" PRIMARY KEY ("numer\_pracownika\_PK")

/

-- Table and Columns comments section

COMMENT ON COLUMN "PracowncyAdministracji"."Poziom" IS 'Mówi o Poziomie Pracownika Administracji.'

/

-- Table PracownicyAdministracji\_Blok

CREATE TABLE "PracownicyAdministracji\_Blok"(

"numer\_bloku\_PK" Integer NOT NULL,

"numer\_pracownika\_PK" Integer NOT NULL

)

/

-- Table PracownicyAdministracji\_DomJednorodzinny

CREATE TABLE "PracownicyAdministracji\_DomJednorodzinny"(

"numer\_domu\_PK" Integer NOT NULL,

"numer\_pracownika\_PK" Integer NOT NULL

)

/

-- Table PracownicySprzatajacy\_Blok

CREATE TABLE "PracownicySprzatajacy\_Blok"(

"numer\_bloku\_PK" Integer NOT NULL,

"numer\_pracownika\_PK" Integer NOT NULL

)

/

-- Table Adresy

CREATE TABLE "Adresy"(

"Nr\_adresu" Integer NOT NULL,

"Miasto" Varchar2(20 ) NOT NULL,

"Ulica" Varchar2(30 ) NOT NULL,

"Nr\_lokalu" Varchar2(4 ) NOT NULL,

"Nr\_poczty" Integer NOT NULL

)

/

-- Create indexes for table Adresy

CREATE INDEX "IX\_Relationship5" 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 (kllucz główny).'

/

COMMENT ON COLUMN "Adresy"."Miasto" IS 'Miasto'

/

COMMENT ON COLUMN "Adresy"."Ulica" IS 'Ulica'

/

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

/

-- Table Administratorzy

CREATE TABLE "Administratorzy"(

"Nr\_administratora" Integer NOT NULL,

"Imie" Varchar2(20 ) NOT NULL,

"Nazwisko" Varchar2(30 ) NOT NULL,

"Rozpoczecie\_pracy" Date NOT NULL,

"Zakonczenie\_pracy" Date,

"numer\_strony\_PK" Integer NOT NULL

)

/

-- Create indexes for table Administratorzy

CREATE INDEX "IX\_Relationship3" ON "Administratorzy" ("numer\_strony\_PK")

/

-- Add keys for table Administratorzy

ALTER TABLE "Administratorzy" ADD CONSTRAINT "PK\_Administratorzy" PRIMARY KEY ("Nr\_administratora")

/

-- Table and Columns comments section

COMMENT ON TABLE "Administratorzy" IS 'Relacja Administratorzy'

/

COMMENT ON COLUMN "Administratorzy"."Nr\_administratora" IS 'Unikatowy identyfikator Administratora.'

/

COMMENT ON COLUMN "Administratorzy"."Imie" IS 'Imię Administratora'

/

COMMENT ON COLUMN "Administratorzy"."Nazwisko" IS 'nazwisko Administratora'

/

COMMENT ON COLUMN "Administratorzy"."Rozpoczecie\_pracy" IS 'Data rozpoczęcia pracy jako Administrator.'

/

COMMENT ON COLUMN "Administratorzy"."Zakonczenie\_pracy" IS 'Data zakończenia pracy jako Administrator'

/

-- Table Poczty

CREATE TABLE "Poczty"(

"Nr\_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 ("Nr\_poczty")

/

ALTER TABLE "Poczty" ADD CONSTRAINT "Kod\_poczty" UNIQUE ("Kod\_poczty")

/

-- Table and Columns comments section

COMMENT ON TABLE "Poczty" IS 'Relacja słownikowa poczty.'

/

COMMENT ON COLUMN "Poczty"."Nr\_poczty" IS 'Unikatowy identyfikator poczty'

/

COMMENT ON COLUMN "Poczty"."Kod\_poczty" IS 'Kod danej poczty '

/

COMMENT ON COLUMN "Poczty"."Poczta" IS 'Lokalizacja poczty'

/

-- Table CzlonkowieZarzadu

CREATE TABLE "CzlonkowieZarzadu"(

"Nr\_czlonka" Integer NOT NULL,

"Imie" Varchar2(20 ) NOT NULL,

"Nazwisko" Varchar2(30 ) NOT NULL,

"numer\_zarzadu\_PK" Integer NOT NULL,

"Nr\_roli" Integer NOT NULL

)

/

-- Create indexes for table CzlonkowieZarzadu

CREATE INDEX "IX\_Relationship6" ON "CzlonkowieZarzadu" ("numer\_zarzadu\_PK")

/

CREATE INDEX "IX\_Relationship9" ON "CzlonkowieZarzadu" ("Nr\_roli")

/

-- Add keys for table CzlonkowieZarzadu

ALTER TABLE "CzlonkowieZarzadu" ADD CONSTRAINT "PK\_CzlonkowieZarzadu" PRIMARY KEY ("Nr\_czlonka")

/

-- Table and Columns comments section

COMMENT ON TABLE "CzlonkowieZarzadu" IS 'Relacja Członkowie Zarządu. '

/

COMMENT ON COLUMN "CzlonkowieZarzadu"."Nr\_czlonka" IS 'Unikatowy identyfikator członka zarządu'

/

COMMENT ON COLUMN "CzlonkowieZarzadu"."Imie" IS 'Imię członka zarządu'

/

COMMENT ON COLUMN "CzlonkowieZarzadu"."Nazwisko" IS 'Nazwisko członka zarządu'

/

-- Table Role

CREATE TABLE "Role"(

"Nr\_roli" Integer NOT NULL,

"Nazwa" Varchar2(20 ) NOT NULL,

"Opis" Varchar2(400 ) NOT NULL

)

/

-- Add keys for table Role

ALTER TABLE "Role" ADD CONSTRAINT "PK\_Role" PRIMARY KEY ("Nr\_roli")

/

-- Table and Columns comments section

COMMENT ON TABLE "Role" IS 'Opisuje pełnioną rolę przez Członka Zarządu.'

/

COMMENT ON COLUMN "Role"."Nr\_roli" IS 'Unikalny identyfikator pełnionej roli.'

/

COMMENT ON COLUMN "Role"."Nazwa" IS 'Nazwa pełnionej roli'

/

COMMENT ON COLUMN "Role"."Opis" IS 'Opis pełnionej roli'

/

-- Table Stanowiska

CREATE TABLE "Stanowiska"(

"Nr\_stanowiska" Integer NOT NULL,

"Nazwa" Varchar2(20 ) NOT NULL,

"Opis" Varchar2(450 ) NOT NULL

)

/

-- Add keys for table Stanowiska

ALTER TABLE "Stanowiska" ADD CONSTRAINT "PK\_Stanowiska" PRIMARY KEY ("Nr\_stanowiska")

/

-- Table and Columns comments section

COMMENT ON COLUMN "Stanowiska"."Nr\_stanowiska" IS 'Unikalny identyfikator stanowiska'

/

COMMENT ON COLUMN "Stanowiska"."Nazwa" IS 'Nazwa pełnionego stanowiska'

/

COMMENT ON COLUMN "Stanowiska"."Opis" IS 'Opis pełnionego stanowiska'

/

-- Table Wynagrodzenia

CREATE TABLE "Wynagrodzenia"(

"Nr\_wynagrodzenia" Integer NOT NULL,

"Data" Date NOT NULL,

"Kwota\_podstawowa" Number(8,2) NOT NULL,

"Kowta\_dodatkowa" Number(8,2),

"numer\_pracownika\_PK" Integer NOT NULL

)

/

-- Create indexes for table Wynagrodzenia

CREATE INDEX "IX\_Relationship12" ON "Wynagrodzenia" ("numer\_pracownika\_PK")

/

-- Add keys for table Wynagrodzenia

ALTER TABLE "Wynagrodzenia" ADD CONSTRAINT "PK\_Wynagrodzenia" PRIMARY KEY ("Nr\_wynagrodzenia")

/

-- Table and Columns comments section

COMMENT ON TABLE "Wynagrodzenia" IS 'Wynagrodzenie pracownika'

/

COMMENT ON COLUMN "Wynagrodzenia"."Nr\_wynagrodzenia" IS 'Unikalny identyfiaktor wynagrodzenia'

/

COMMENT ON COLUMN "Wynagrodzenia"."Data" IS 'Data wypłaty wynagrodzenia dla pracownika'

/

COMMENT ON COLUMN "Wynagrodzenia"."Kwota\_podstawowa" IS 'Wartosć wypłacanego wynagrodzenia podstawowego'

/

COMMENT ON COLUMN "Wynagrodzenia"."Kowta\_dodatkowa" IS 'Wartość wypłacanej kwoty dodatkowej'

/

-- Table DaneNiejawne

CREATE TABLE "DaneNiejawne"(

"Nr\_uprawnien" Integer NOT NULL,

"Data\_waznosci" Date NOT NULL,

"Opis" Varchar2(450 ) NOT NULL,

"numer\_pracownika\_PK" Integer NOT NULL

)

/

-- Create indexes for table DaneNiejawne

CREATE INDEX "IX\_Relationship13" ON "DaneNiejawne" ("numer\_pracownika\_PK")

/

-- Add keys for table DaneNiejawne

ALTER TABLE "DaneNiejawne" ADD CONSTRAINT "PK\_DaneNiejawne" PRIMARY KEY ("Nr\_uprawnien")

/

-- Table and Columns comments section

COMMENT ON TABLE "DaneNiejawne" IS 'Relacja opisująca Dane Niejawne (finansowe) niedostępne dla pozostałych pracowników.'

/

COMMENT ON COLUMN "DaneNiejawne"."Nr\_uprawnien" IS 'Unikatowy identyfikator upranień do danych niejawnych Księgowego'

/

COMMENT ON COLUMN "DaneNiejawne"."Data\_waznosci" IS 'Data ważności uzyskanych uprawnień'

/

COMMENT ON COLUMN "DaneNiejawne"."Opis" IS 'Opis uzyskanych upranwnień, do jakich danych ma dostęp (finansowych)'

/

-- Table Umowy

CREATE TABLE "Umowy"(

"Nr\_umowy" Integer NOT NULL,

"Data\_rozpoczecia" Date NOT NULL,

"Data\_zakonczenia" Date,

"Opis" Varchar2(800 ) NOT NULL

)

/

-- Add keys for table Umowy

ALTER TABLE "Umowy" ADD CONSTRAINT "PK\_Umowy" PRIMARY KEY ("Nr\_umowy")

/

-- Table and Columns comments section

COMMENT ON COLUMN "Umowy"."Nr\_umowy" IS 'Unikatowy numer identyfikujący umowę'

/

COMMENT ON COLUMN "Umowy"."Data\_rozpoczecia" IS 'Data wejścia umowy w życie'

/

COMMENT ON COLUMN "Umowy"."Data\_zakonczenia" IS 'Data zakończenia obowiązywania umowy'

/

COMMENT ON COLUMN "Umowy"."Opis" IS 'Treść umowy, precyzująca jej rodzaj, formę, warunki itp.'

/

-- Trigger for sequence NrSpoldzielniSeq for column numer\_spoldzielni\_PK in table SpoldzielnieMieszkaniowe ---------

CREATE OR REPLACE TRIGGER "ts\_SpoldzielnieMieszkaniowe\_NrSpoldzielniSeq" BEFORE INSERT

ON "SpoldzielnieMieszkaniowe" FOR EACH ROW

BEGIN

:new."numer\_spoldzielni\_PK" := "NrSpoldzielniSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_SpoldzielnieMieszkaniowe\_NrSpoldzielniSeq" AFTER UPDATE OF "numer\_spoldzielni\_PK"

ON "SpoldzielnieMieszkaniowe" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "numer\_spoldzielni\_PK" in table "SpoldzielnieMieszkaniowe" as it uses sequence.');

END;

/

-- Trigger for sequence NrZarzaduSeq for column numer\_zarzadu\_PK in table Zarzady ---------

CREATE OR REPLACE TRIGGER "ts\_Zarzady\_NrZarzaduSeq" BEFORE INSERT

ON "Zarzady" FOR EACH ROW

BEGIN

:new."numer\_zarzadu\_PK" := "NrZarzaduSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_Zarzady\_NrZarzaduSeq" AFTER UPDATE OF "numer\_zarzadu\_PK"

ON "Zarzady" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "numer\_zarzadu\_PK" in table "Zarzady" as it uses sequence.');

END;

/

-- Trigger for sequence NrBlokuSeq for column numer\_bloku\_PK in table Bloki ---------

CREATE OR REPLACE TRIGGER "ts\_Bloki\_NrBlokuSeq" BEFORE INSERT

ON "Bloki" FOR EACH ROW

BEGIN

:new."numer\_bloku\_PK" := "NrBlokuSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_Bloki\_NrBlokuSeq" AFTER UPDATE OF "numer\_bloku\_PK"

ON "Bloki" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "numer\_bloku\_PK" in table "Bloki" as it uses sequence.');

END;

/

-- Trigger for sequence NrMieszkaniaSeq for column numer\_mieszkania\_PK in table Mieszkania ---------

CREATE OR REPLACE TRIGGER "ts\_Mieszkania\_NrMieszkaniaSeq" BEFORE INSERT

ON "Mieszkania" FOR EACH ROW

BEGIN

:new."numer\_mieszkania\_PK" := "NrMieszkaniaSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_Mieszkania\_NrMieszkaniaSeq" AFTER UPDATE OF "numer\_mieszkania\_PK"

ON "Mieszkania" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "numer\_mieszkania\_PK" in table "Mieszkania" as it uses sequence.');

END;

/

-- Trigger for sequence NrDzialkiSeq for column numer\_dzialki\_PK in table DzialkiBudowlane ---------

CREATE OR REPLACE TRIGGER "ts\_DzialkiBudowlane\_NrDzialkiSeq" BEFORE INSERT

ON "DzialkiBudowlane" FOR EACH ROW

BEGIN

:new."numer\_dzialki\_PK" := "NrDzialkiSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_DzialkiBudowlane\_NrDzialkiSeq" AFTER UPDATE OF "numer\_dzialki\_PK"

ON "DzialkiBudowlane" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "numer\_dzialki\_PK" in table "DzialkiBudowlane" as it uses sequence.');

END;

/

-- Trigger for sequence NrPracownikaSeq for column numer\_pracownika\_PK in table Pracownicy ---------

CREATE OR REPLACE TRIGGER "ts\_Pracownicy\_NrPracownikaSeq" BEFORE INSERT

ON "Pracownicy" FOR EACH ROW

BEGIN

:new."numer\_pracownika\_PK" := "NrPracownikaSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_Pracownicy\_NrPracownikaSeq" AFTER UPDATE OF "numer\_pracownika\_PK"

ON "Pracownicy" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "numer\_pracownika\_PK" in table "Pracownicy" as it uses sequence.');

END;

/

-- Trigger for sequence NrKlientaSeq for column numer\_klienta\_PK in table Klienci ---------

CREATE OR REPLACE TRIGGER "ts\_Klienci\_NrKlientaSeq" BEFORE INSERT

ON "Klienci" FOR EACH ROW

BEGIN

:new."numer\_klienta\_PK" := "NrKlientaSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_Klienci\_NrKlientaSeq" AFTER UPDATE OF "numer\_klienta\_PK"

ON "Klienci" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "numer\_klienta\_PK" in table "Klienci" as it uses sequence.');

END;

/

-- Trigger for sequence NrStronySeq for column numer\_strony\_PK in table StronyInternetowe ---------

CREATE OR REPLACE TRIGGER "ts\_StronyInternetowe\_NrStronySeq" BEFORE INSERT

ON "StronyInternetowe" FOR EACH ROW

BEGIN

:new."numer\_strony\_PK" := "NrStronySeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_StronyInternetowe\_NrStronySeq" AFTER UPDATE OF "numer\_strony\_PK"

ON "StronyInternetowe" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "numer\_strony\_PK" in table "StronyInternetowe" as it uses sequence.');

END;

/

-- Trigger for sequence NrDomuSeq for column numer\_domu\_PK in table DomyJednorodzinne ---------

CREATE OR REPLACE TRIGGER "ts\_DomyJednorodzinne\_NrDomuSeq" BEFORE INSERT

ON "DomyJednorodzinne" FOR EACH ROW

BEGIN

:new."numer\_domu\_PK" := "NrDomuSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_DomyJednorodzinne\_NrDomuSeq" AFTER UPDATE OF "numer\_domu\_PK"

ON "DomyJednorodzinne" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "numer\_domu\_PK" in table "DomyJednorodzinne" as it uses sequence.');

END;

/

-- Trigger for sequence NrAdresuSeq for column Nr\_adresu in table Adresy ---------

CREATE OR REPLACE TRIGGER "ts\_Adresy\_NrAdresuSeq" BEFORE INSERT

ON "Adresy" FOR EACH ROW

BEGIN

:new."Nr\_adresu" := "NrAdresuSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_Adresy\_NrAdresuSeq" 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 NrAdministratoraSeq for column Nr\_administratora in table Administratorzy ---------

CREATE OR REPLACE TRIGGER "ts\_Administratorzy\_NrAdministratoraSeq" BEFORE INSERT

ON "Administratorzy" FOR EACH ROW

BEGIN

:new."Nr\_administratora" := "NrAdministratoraSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_Administratorzy\_NrAdministratoraSeq" AFTER UPDATE OF "Nr\_administratora"

ON "Administratorzy" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "Nr\_administratora" in table "Administratorzy" as it uses sequence.');

END;

/

-- Trigger for sequence NrPocztySeq for column Nr\_poczty in table Poczty ---------

CREATE OR REPLACE TRIGGER "ts\_Poczty\_NrPocztySeq" BEFORE INSERT

ON "Poczty" FOR EACH ROW

BEGIN

:new."Nr\_poczty" := "NrPocztySeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_Poczty\_NrPocztySeq" 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 NrCzlonkaSeq for column Nr\_czlonka in table CzlonkowieZarzadu ---------

CREATE OR REPLACE TRIGGER "ts\_CzlonkowieZarzadu\_NrCzlonkaSeq" BEFORE INSERT

ON "CzlonkowieZarzadu" FOR EACH ROW

BEGIN

:new."Nr\_czlonka" := "NrCzlonkaSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_CzlonkowieZarzadu\_NrCzlonkaSeq" AFTER UPDATE OF "Nr\_czlonka"

ON "CzlonkowieZarzadu" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "Nr\_czlonka" in table "CzlonkowieZarzadu" as it uses sequence.');

END;

/

-- Trigger for sequence NrRoliSeq for column Nr\_roli in table Role ---------

CREATE OR REPLACE TRIGGER "ts\_Role\_NrRoliSeq" BEFORE INSERT

ON "Role" FOR EACH ROW

BEGIN

:new."Nr\_roli" := "NrRoliSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_Role\_NrRoliSeq" AFTER UPDATE OF "Nr\_roli"

ON "Role" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "Nr\_roli" in table "Role" as it uses sequence.');

END;

/

-- Trigger for sequence NrUprawnienSeq for column Nr\_uprawnien in table DaneNiejawne ---------

CREATE OR REPLACE TRIGGER "ts\_DaneNiejawne\_NrUprawnienSeq" BEFORE INSERT

ON "DaneNiejawne" FOR EACH ROW

BEGIN

:new."Nr\_uprawnien" := "NrUprawnienSeq".nextval;

END;

/

CREATE OR REPLACE TRIGGER "tsu\_DaneNiejawne\_NrUprawnienSeq" AFTER UPDATE OF "Nr\_uprawnien"

ON "DaneNiejawne" FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20010,'Cannot update column "Nr\_uprawnien" in table "DaneNiejawne" as it uses sequence.');

END;

/

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

ALTER TABLE "Zarzady" ADD CONSTRAINT "Jest\_zarzadzana\_przez" FOREIGN KEY ("numer\_spoldzielni\_PK") REFERENCES "SpoldzielnieMieszkaniowe" ("numer\_spoldzielni\_PK")

/

ALTER TABLE "StronyInternetowe" ADD CONSTRAINT "Posiada\_strone" FOREIGN KEY ("numer\_spoldzielni\_PK") REFERENCES "SpoldzielnieMieszkaniowe" ("numer\_spoldzielni\_PK")

/

ALTER TABLE "Klienci" ADD CONSTRAINT "Zamieszkuje\_mieszkanie" FOREIGN KEY ("numer\_mieszkania\_PK") REFERENCES "Mieszkania" ("numer\_mieszkania\_PK")

/

ALTER TABLE "DzialkiBudowlane" ADD CONSTRAINT "Posiada\_dzialke" FOREIGN KEY ("numer\_spoldzielni\_PK") REFERENCES "SpoldzielnieMieszkaniowe" ("numer\_spoldzielni\_PK")

/

ALTER TABLE "DomyJednorodzinne" ADD CONSTRAINT "Posiada\_dom" FOREIGN KEY ("numer\_spoldzielni\_PK") REFERENCES "SpoldzielnieMieszkaniowe" ("numer\_spoldzielni\_PK")

/

ALTER TABLE "Pracownicy" ADD CONSTRAINT "Zatrudnia" FOREIGN KEY ("numer\_spoldzielni\_PK") REFERENCES "SpoldzielnieMieszkaniowe" ("numer\_spoldzielni\_PK")

/

ALTER TABLE "Mieszkania" ADD CONSTRAINT "Zawiera\_mieszkania" FOREIGN KEY ("numer\_bloku\_PK") REFERENCES "Bloki" ("numer\_bloku\_PK")

/

ALTER TABLE "Bloki" ADD CONSTRAINT "Posiada\_blok" FOREIGN KEY ("numer\_spoldzielni\_PK") REFERENCES "SpoldzielnieMieszkaniowe" ("numer\_spoldzielni\_PK")

/

ALTER TABLE "Klienci" ADD CONSTRAINT "Zameszkuje\_Dom" FOREIGN KEY ("numer\_domu\_PK") REFERENCES "DomyJednorodzinne" ("numer\_domu\_PK")

/

ALTER TABLE "Klienci" ADD CONSTRAINT "Posiada\_umowe" FOREIGN KEY ("numer\_spoldzielni\_PK") REFERENCES "SpoldzielnieMieszkaniowe" ("numer\_spoldzielni\_PK")

/

ALTER TABLE "Klienci" ADD CONSTRAINT "Jest\_zamieszkiwany" FOREIGN KEY ("numer\_bloku\_PK") REFERENCES "Bloki" ("numer\_bloku\_PK")

/

ALTER TABLE "Administratorzy" ADD CONSTRAINT "Posiada\_administratora" FOREIGN KEY ("numer\_strony\_PK") REFERENCES "StronyInternetowe" ("numer\_strony\_PK")

/

ALTER TABLE "SpoldzielnieMieszkaniowe" ADD CONSTRAINT "Spoldzielnia\_ma\_adres" FOREIGN KEY ("Nr\_adresu") REFERENCES "Adresy" ("Nr\_adresu")

/

ALTER TABLE "Adresy" ADD CONSTRAINT "Adres\_posiada\_poczte" FOREIGN KEY ("Nr\_poczty") REFERENCES "Poczty" ("Nr\_poczty")

/

ALTER TABLE "CzlonkowieZarzadu" ADD CONSTRAINT "Posiada\_czlonkow" FOREIGN KEY ("numer\_zarzadu\_PK") REFERENCES "Zarzady" ("numer\_zarzadu\_PK")

/

ALTER TABLE "CzlonkowieZarzadu" ADD CONSTRAINT "Pelni\_role" FOREIGN KEY ("Nr\_roli") REFERENCES "Role" ("Nr\_roli")

/

ALTER TABLE "Pracownicy" ADD CONSTRAINT "Pracownik\_posiada\_stanowisko" FOREIGN KEY ("Nr\_stanowiska") REFERENCES "Stanowiska" ("Nr\_stanowiska")

/

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

/

ALTER TABLE "Wynagrodzenia" ADD CONSTRAINT "Pracownik\_posiada\_wynagrodzenie" FOREIGN KEY ("numer\_pracownika\_PK") REFERENCES "Pracownicy" ("numer\_pracownika\_PK")

/

ALTER TABLE "DaneNiejawne" ADD CONSTRAINT "PracownikKsiegowosci\_posiada\_dostep\_do\_danych\_niejawnych" FOREIGN KEY ("numer\_pracownika\_PK") REFERENCES "PracownicyKsiegowosci" ("numer\_pracownika\_PK")

/

ALTER TABLE "DzialkiBudowlane" ADD CONSTRAINT "DzialkaBudowlana\_posiada\_adres" FOREIGN KEY ("Nr\_adresu") REFERENCES "Adresy" ("Nr\_adresu")

/

ALTER TABLE "DomyJednorodzinne" ADD CONSTRAINT "DomJednorodzinny\_posiada\_adres" FOREIGN KEY ("Nr\_adresu") REFERENCES "Adresy" ("Nr\_adresu")

/

ALTER TABLE "Bloki" ADD CONSTRAINT "Blok\_posiada\_adres" FOREIGN KEY ("Nr\_adresu") REFERENCES "Adresy" ("Nr\_adresu")

/

ALTER TABLE "Klienci" ADD CONSTRAINT "Klient\_posiada\_adres" FOREIGN KEY ("Nr\_adresu") REFERENCES "Adresy" ("Nr\_adresu")

/

ALTER TABLE "Pracownicy" ADD CONSTRAINT "Pracownik\_posiada\_umowe" FOREIGN KEY ("Nr\_umowy") REFERENCES "Umowy" ("Nr\_umowy")

/

ALTER TABLE "Klienci" ADD CONSTRAINT "Klient\_posiada\_umowe" FOREIGN KEY ("Nr\_umowy") REFERENCES "Umowy" ("Nr\_umowy")

/

# **Bibliografia**

* Wykład i materiały wykładowe z przedmiotu „Bazy Danych i Big Data” prowadzonego przez dr hab. inż. Marcina Kowalczyka w semestrze zimowym 2021
* Dokumentacja programów Oracle19c, SQL Developer oraz TOAD Data Modeller