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** 2](#_Toc88668052)

[**2.** **Definicja systemu** 3](#_Toc88668053)

[**3.** **Model konceptualny** 5](#_Toc88668054)

[**4.** **Model logiczny** 19](#_Toc88668055)

[**4.1. Usunięcie właściwości niekompatybilnych z modelem relacyjnym** 19](#_Toc88668056)

[**4.2. Proces normalizacji** 20](#_Toc88668057)

[**4.3. Schemat ER na poziomie modelu logicznego** 22](#_Toc88668058)

[**4.4. Więzy integralności** 22](#_Toc88668059)

[**4.5. Proces denormalizacji** 23](#_Toc88668060)

[**5.** **Model fizyczny** 24](#_Toc88668061)

[**6.** **Bibliografia** 25](#_Toc88668062)

# **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
* 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**

SpoldzielniaMieszkaniowa

Określa strukturę danych i reprezentuje spółdzielnie mieszkaniową, jako główną encję “wyjściową”.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_spoldzielni\_PK | numer\_spoldzielni\_PK | SmallInt | - | yes | yes | Numer jednoznacznie identyfikujacy spoldzielnie |
| adres\_spoldzielni | adres\_spoldzielni | VarChar | - | no | yes | Reprezentuje adres spoldzielni - pole segmentowa (miasto, kod pocztowy, ulica, numer budynku, numer lokalu) |
| REGON | REGON | Char | 10 | no | yes | Reprezentuje numer REGON spoldzielni - klucz kandydujacy |
| KRS | KRS | Char | 9 | no | yes | Reprezentuje nume KRS spoldzielni - klucz kandydujacy |
| NIP | NIP | Char | 9 | no | yes | Reprezentuje numer NIP spoldzielni - klucz kandydujacy |

Zarzad

Określa strukturę danych i reprezentuje zarząd spółdzielni mieszkaniowej.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_zarzadu\_pk | numer\_zarzadu\_PK | SmallInt | - | yes | yes | Reprezentuje numer jednoznacznie identyfikujacy zarzad - klucz glowny |
| skarbnik | skarbnik | VarChar |  | no | yes | Reprezentuje skarbnika zarzadu |
| prezes | prezes | VarChar |  | no | yes | Reprezentuje prezesa zarzadu |
| vice prezes | vice prezes | VarChar |  | no | yes | Reprezentuje vice prezesa zarzadu |
| rozpoczecie\_kadencji | rozpoczecie\_kadencji | Date |  | no | yes | Reprezentuje date, kiedy zarzad rozpoczal swoja kadencje |
| planowane\_zakonczenie\_kadencji | planowane\_zakonczenie\_kadencji | Date |  | no | yes | Reprezentuje planowana date zakonczenia kadencji end |

Klient

Określa strukturę danych i reprezentuje klienta spółdzielni mieszkaniowej .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_klienta\_PK | numer\_klienta\_PK | SmallInt | - | yes | yes | Reprezentuje numer jednoznacznie identyfikujacy klienta spoldzielni - klucz glowny |
| imie | imie | VarChar | 64 | no | yes | Reprezentuje imie klienta |
| nazwisko | nazwisko | VarChar | 64 | no | yes | Reprezentuje nazwisko klienta |
| plec | plec | Char | 1 | no | yes | Reprezentuje plec klienta |
| adres\_klienta | adres\_klienta | VarChar | 512 | no | yes | Reprezentuje adres klienta |
| PESEL | PESEL | Character | 11 | no | no | Reprezentuje adres klienta |
| data\_urodzenia | data\_urodzenia | Date | - | no | no | Reprezentuje date urodzenia klienta |
| od\_kiedy\_klient | od\_kiedy\_klient | Date | - | no | no | Reprezentuje date od ktorej klient korzysta z uslug spoldzielni |
| numer\_telefonu | numer\_telefonu | VarChar | 12 | no | no | Reprezentuje numer telefonu klienta |

Pracownik

Określa strukturę danych i reprezentuje encję Pracownika spółdzielni.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_pracownika\_PK | numer\_pracownika\_PK | SmallInt | - | yes | yes | Reprezentuje numer jednoznacznie identyfikujacy pracownika - klucz glowny |
| pensja\_miesieczna | pensja\_miesieczna | Money | - | no | yes | Reprezentuje miesieczna pensje pracownika |
| umowa\_podpisana | umowa\_podpisana | Date | - | no | yes | Reprezentuje date podpisania umowy z pracownikiem |
| plec | plec | PlecD | - | no | yes | Reprezentuje plec pracownika |
| stanowisko | stanowisko | VarChar | 512 | no | yes | Reprezentuje stanowisko, na jakim pracuje pracownik |
| adres\_pracownika | adres\_pracownika | AdresD | - | no | yes | Reprezentuje adres zamieszkania pracownika |
| umowa\_rozwiazana | umowa\_rozwiazana | Date | - | no | no | Reprezentuje date rozwiazania umowy z pracownikiem |
| PESEL | PESEL | Character | 10 | no | no | Reprezentuje numer PESEL pracownika |
| numer\_telefonu | numer\_telefonu | VarChar | 12 | no | no | Reprezentuje numer telefonu pracownika |
| data\_urodzenia | data\_urodzenia | Date | - | no | no | Reprezentuje date urodzenia pracownika |
| email\_pracownika | email\_pracownika | VarChar | 512 | no | no | Reprezentuje adres email pracownika |
| imie | imie | VarChar | 64 | no | yes | Reprezentuje imie pracownika |
| nazwisko | nazwisko | VarChar | 64 | no | yes | Reprezentuje nazwisko pracownika |

PracownikKsiegowosci

Określa strukturę danych i reprezentuje encję Pracownika Księgowości.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_pracownika\_PK | numer\_pracownika\_PK | SmallInt | - | yes | yes | Reprezentuje numer jednoznacznie identyfikujacy pracownika - klucz glowny |
| dostep\_do\_danych\_niejawnych | dostep\_do\_danych\_niejawnych | VarChar | 512 | no | yes | Reprezentuje pracownika ksiegowosci spoldzielni mieszkaniowej |
| numer\_uprawnien | numer\_uprawnien | VarChar | 64 | no | yes | Reprezentuje numer uprawnień posiadanych przez pracownika ksiegowego |

PracownikSprzątający

Określa strukturę danych i reprezentuje encję Pracownika Sprzątającego w spółdzielni.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_pracownika\_PK | numer\_pracownika\_PK | SmallInt | - | yes | yes | jak wyżej |
| sprzatne\_bloki | sprzatne\_bloki | VarChar | 512 | no | yes | Reprezentuje bloki przydzielone do sprzatania pracownikowi |

PracownikAministracji

Określa strukturę danych i reprezentuje encję Pracownika Administracji, który jest zatrudniony przez spółdzielnię i odpowiada za zarządzanie posiadanymi przez nie nieruchomościami

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_pracownika\_PK | numer\_pracownika\_PK | SmallInt | - | yes | yes | jak wyżej |
| zarzadzane\_bloki | zarzadzane\_bloki | VarChar | 512 | no | yes | Reprezentuje bloki zarzadzane przez administratora nieruchomosci |
| zarzadzne\_domy | zarzadzne\_domy | VarChar | 512 | no | yes | Reprezentuje domy zarzadzane przez administratora nieruchomosci |
|  |  |  |  |  |  |  |

StronaInternetowa

Określa strukturę danych i reprezentuje encję Strony Internetowej, nalezącej do spółdzielni.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_strony\_PK | numer\_strony\_PK | SmallInt | - | yes | yes | Reprezentuje numer jednoznacznie identyfikujacy strone internetowa - klucz glowny |
| administrator | administrator | VarChar | 256 | no | yes | Reprezentuje administratora strony |
| adres\_url | adres\_url | VarChar | 256 | no | yes | Reprezentuje adres URL strony |

DzialkaBudowlana

Określa strukturę danych i reprezentuje encję Działki Budowlanej należącej do spółdzielni, na której powstaje nowy blok/dom.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_dzialki\_PK | numer\_dzialki\_PK | SmallInt | - | yes | yes | Reprezentuje numer jednoznacznie identyfikujacy dzialke (nie numer w sensie adresu!) - klucz glowny |
| powierzchnia | powierzchnia | Float | 10 | no | no | Reprezentuje powierzchnie dzialki |
| adres\_dzialki | adres\_dzialki | AdresD | - | no | yes | Reprezentuje adres dzialki - pole segmentowe |
| media | media | MediaD | - | no | no | Reprezentuje media dostepne na dzialce (gaz, prad, woda) - pole wyliczeniowe |
| garaz | garaz | Boolean | - | no | no | Reprezentuje wiedze czy na dzialce znajduje sie garaz |
| ksiega\_wieczysta | ksiega\_wieczysta | Boolean | - | no | yes | Reprezentuje wiedze czy dzialka aktualny wpis do ksiegi wieczystej |

DomJednorodzinny

Określa strukturę danych i reprezentuje Dom Jednorodzinny należący do spółdzielni.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_domu\_PK | numer\_domu\_PK | SmallInt | - | yes | yes | Reprezentuje numer jednoznacznie identyfikujacy dom (nie numer domu w sensie adresu!) - klucz glowny |
| powierzchnia | powierzchnia | Float | 2 | no | no | Reprezentuje powierzchnie domu |
| adres\_domu | adres\_domu | AdresD | - | no | yes | Reprezentuje adres domu |
| ilosc\_pieter | ilosc\_pieter | SmallInt | - | no | no | Reprezentuje ilosc pieter domu |
| budynki\_gospodarcze | budynki\_gospodarcze | VarChar | 1024 | no | no | Reprezentuje wiedze na temat innych budynkow znajdujacych sie na dzialce z domem (garaz, komorka ogrodnicza, itd.) |
| ilosc\_pokoi | ilosc\_pokoi | SmallInt | - | no | no | Reprezentuje ilosc pokoi (innych niz lazienka) znajdujacych sie w domu |
| ilosc\_lazienek | ilosc\_lazienek | SmallInt | - | no | no | Reprezentuje ilosc lazienek znajdujacych sie w domu |
|  |  |  |  |  |  |  |

Blok

Określa strukturę danych i reprezentuje encję Blok, należący do spółdzielni.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_bloku\_PK | numer\_bloku\_PK | SmallInt | - | yes | yes | Reprezentuje numer jednoznacznie identyfikujacy blok (nie numer bloku w sensie adresu!) - klucz glowny |
| ilosc\_mieszkan | ilosc\_mieszkan | SmallInt | - | no | no | Reprezentuje ilosc mieszkan |
| adres\_bloku | adres\_bloku | AdresD | - | no | yes | Reprezentuje adres bloku |

Mieszkanie

Określa strukturę danych i reprezentuje encję Mieszkanie, należące do spółdzielnie, a zawierające się w bloku.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Caption** | **Name** | **DataType** | **DataType param** | **PK** | **Mandatory** | **Description** |
| numer\_mieszkania\_PK | numer\_mieszkania\_PK | SmallInt | - | yes | yes | Reprezentuje numer jednoznacznie identyfikujacy mieszkanie (nie numer mieszkania w sensie adresu!) - klucz glowny |
| ilosc\_pokoi | ilosc\_pokoi | SmallInt | - | no | no | Reprezentuje ilosc pokoi znajdujacych sie w mieszkaniu |
| ilosc\_lazienek | ilosc\_lazienek | SmallInt | - | no | no | Reprezentuje ilosc lazienek znajdujacych sie w mieszkaniu |

**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.

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

**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 |

**3.4. Schemat ER na poziomie konceptualnym**

Zobacz: zal\_1\_model\_konceptualny.png

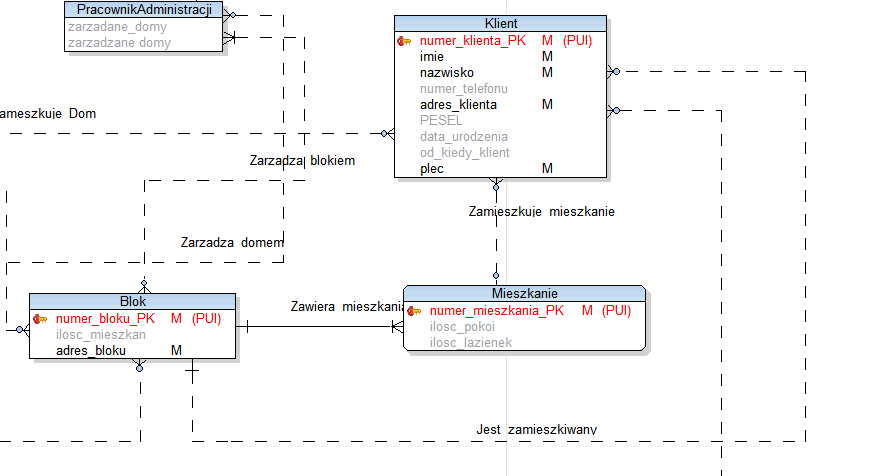
**3.5. Pułapki szczelinowe i wachlarzowe**

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**

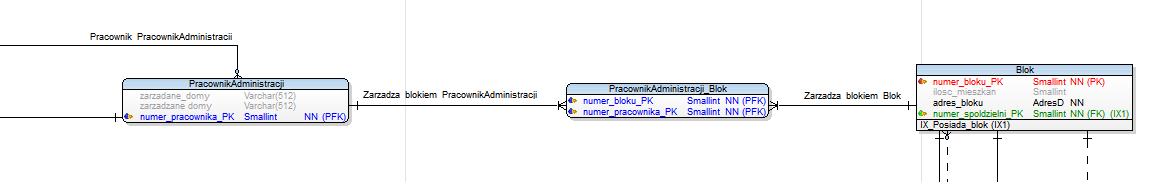
# **4.1. Usunięcie właściwości niekompatybilnych z modelem relacyjnym**

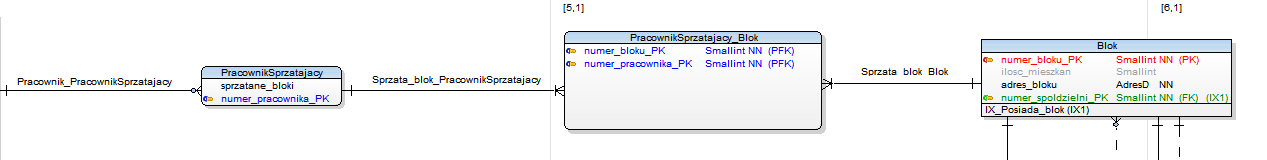
Aby przejść od zaproponowanego modelu konceptualnego do logicznego modelu relacyjnego podjęto próbę usunięcia niekompatybilności modelu konceptualnego z modelem relacyjnym poprzez następujące działania:

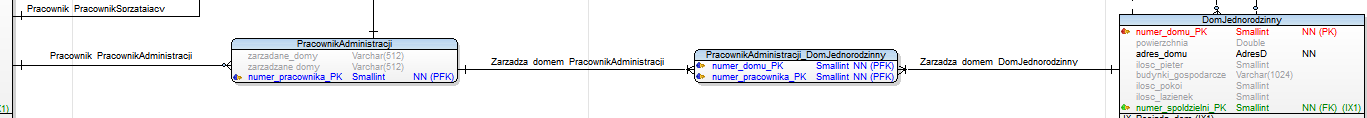
* związki wiele do wielu zastąpiono tzw. tablicami “bridge’ującymi”
* nazwa każdej encji została zmieniona na liczbę mnogą w celu odróżnienia relacji od encji
* identyfikujące atrybuty stały się kluczami głównymi tabeli , pozostałe atrybuty stały się niegłównymi atrybutami tabeli

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**

4.2.1. 1PN

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.

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.

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 usunięta atrybut ‘Administratorzy’ z pierwotnej relacji.

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”.

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”.

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.

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.

Stanowiska:

Relacja w takiej postaci jak pokazano spełnia już warunki 2PN oraz 3PN.

Wynagrodzenia:

Relacja w takiej postaci jak pokazano spełnia już warunki 2PN oraz 3PN.

Umowy:

Relacja w takiej postaci jak pokazano spełnia już warunki 2PN oraz 3PN.

Specjalizacje pracowników:

***PracownicyKsięgowści:***

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

***PracownicyAdministracji:***

Dodałem analogicznie jak dla Księgowości atrybut Poziom, a te dwa “zarządzane\_domy” i “zarzadzane\_bloki” to chyba można wywalić, skoro i tak potem idą te bridge do bloków i domów

***PracownicySprzątajacy:***

Tutaj jak wyżej chyba tez można wyrzucić ten atrybut “sprzatane\_bloki” bo sa tez tablice bridge’ujące

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”.

DziałkiBudowlane:

Relacja jest w 3PN.

DomyJednorodzinne:

Relacja jest w 3PN.

Bloki:

Relacja jest w 3PN.

Mieszkania:

Relacja jest w 3PN.

4.2.2. 2PN

4.2.3. 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**

**5.1. Projekt transakcji i weryfikacja ich wykonalności**

**5.2. Strojenie bazy danych poprzez dobór indeksów**

**5.3. Skrypt SQL generujący bazę danych**

**5.4. Przykłady zapytań i poleceń SQL odnoszących się do bazy dancy**

# **Bibliografia**

# 