

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
2. Definicja systemu	3
3. Model konceptualny.....	5
4. Model logiczny	19
4.1. Usunięcie właściwości niekompatybilnych z modelem relacyjnym	19
4.2. Proces normalizacji	20
4.3. Schemat ER na poziomie modelu logicznego.....	22
4.4. Więzy integralności.....	22
4.5. Proces denormalizacji	23
5. Model fizyczny.....	24
6. Bibliografia	25

1. Zakres i cel projektu

Celem pierwszej części projektu jest zaprojektowanie oraz zaimplementowanie relacyjnej bazy danych na poziomach conceptualnym 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.

2. 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.

3. Model konceptualny

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

SpoldzielniaMieszkaniowa

Określa strukturę danych i reprezentuje spółdzielnię mieszkaniową, jako główną encję "wyjściową".

Caption	Name	DataType	DataType param	PK	Mandatory	Description
numer_spoldzielni_PK	numer_spoldzielni_PK	SmallInt	-	yes	yes	Numer jednoznacznie identyfikujący spółdzielnię
adres_spoldzielni	adres_spoldzielni	VarChar	-	no	yes	Reprezentuje adres spółdzielni - pole segmentowa (miasto, kod pocztowy, ulica, numer budynku, numer lokalu)
REGON	REGON	Char	10	no	yes	Reprezentuje numer REGON spółdzielni - klucz kandydujący
KRS	KRS	Char	9	no	yes	Reprezentuje numer KRS spółdzielni - klucz kandydujący

NIP	NIP	Char	9	no	yes	Reprezentuj e numer NIP spoldzielni - klucz kandydujacy
-----	-----	------	---	----	-----	--

Zarząd

Określa strukturę danych i reprezentuje zarząd spółdzielni mieszkaniowej.

Caption	Name	DataType	DataType param	PK	Mandatory	Description
numer_zar zadu_pk	numer_zar zadu_PK	SmallInt	-	yes	yes	Reprezentuj e numer jednoznacz nie identyfikujac y zarząd - klucz glówny
skarbnik	skarbnik	VarChar		no	yes	Reprezentuj e skarbnika zarządu
prezes	prezes	VarChar		no	yes	Reprezentuj e prezesa zarządu
vice prezes	vice prezes	VarChar		no	yes	Reprezentuj e vice prezesa zarządu
rozpoczec ie_kadenc ji	rozpoczec ie_kadenc ji	Date		no	yes	Reprezentuj e date, kiedy zarząd rozpoczał swoja kadencje
planowan e_zakonc zenie_kad encji	planowan e_zakonc zenie_kad encji	Date		no	yes	Reprezentuj e planowana date

						zakonczenia kadencji end
--	--	--	--	--	--	--------------------------------

Klient

Określa strukturę danych i reprezentuje klienta spółdzielni mieszkaniowej .

Caption	Name	DataType	DataType param	PK	Mandatory	Description
numer_kli enta_PK	numer_kli enta_PK	SmallInt	-	yes	yes	Reprezentuj e numer jednoznacz nie identyfikujac y klienta spoldzielni - klucz glowny
imie	imie	VarChar	64	no	yes	Reprezentuj e imie klienta
nazwisko	nazwisko	VarChar	64	no	yes	Reprezentuj e nazwisko klienta
plec	plec	Char	1	no	yes	Reprezentuj e plec klienta
adres_klie nta	adres_klie nta	VarChar	512	no	yes	Reprezentuj e adres klienta
PESEL	PESEL	Character	11	no	no	Reprezentuj e adres klienta
data_urod zenia	data_urod zenia	Date	-	no	no	Reprezentuj e date urodzenia klienta
od_kiedy_ klient	od_kiedy_ klient	Date	-	no	no	Reprezentuj e date od ktorej klient

						korzysta z usług spółdzielni
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 identyfikujący pracownika - klucz główny
pensja_miesieczna	pensja_miesieczna	Money	-	no	yes	Reprezentuje miesięczną pensję pracownika
umowa_podpisana	umowa_podpisana	Date	-	no	yes	Reprezentuje datę podpisania umowy z pracownikiem
plec	plec	PlecD	-	no	yes	Reprezentuje płeć 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_rozwiazania	umowa_rozwiazania	Date	-	no	no	Reprezentuje date rozwiązania 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_pr	numer_pr	SmallInt	-	yes	yes	Reprezentuj

acownika _PK	acownika _PK					e numer jednoznacz nie identyfikujac y pracownika - klucz glowny
dostep_d o_danych _niejawny ch	dostep_d o_danych _niejawny ch	VarChar	512	no	yes	Reprezentuj e pracownika ksiegowosci spoldzielni mieszkanio wej
numer_up rawnien	numer_up rawnien	VarChar	64	no	yes	Reprezentuj e numer uprawnien posiadanych h przez pracownika ksiegowego

PracownikSprzatajacy

Określa strukturę danych i reprezentuje encję Pracownika Sprzątającego w spółdzielni.

Caption	Name	DataType	DataType param	PK	Mandatory	Description
numer_pr acownika _PK	numer_pr acownika _PK	SmallInt	-	yes	yes	jak wyżej
sprzatne_ bloki	sprzatne_ bloki	VarChar	512	no	yes	Reprezentuj e bloki przydzielon e do sprzatania pracowniko wi

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 zarządzane przez administratora nieruchomości
zarzadzane_domy	zarzadzane_domy	VarChar	512	no	yes	Reprezentuje domy zarządzane przez administratora nieruchomości

StronaInternetowa

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

Caption	Name	DataType	DataType param	PK	Mandatory	Description
numer_strony_PK	numer_strony_PK	SmallInt	-	yes	yes	Reprezentuje numer jednoznacznie identyfikujący stronę internetową

						- klucz glówny
administr ator	administr ator	VarChar	256	no	yes	Reprezentuj e administrato ra strony
adres_url	adres_url	VarChar	256	no	yes	Reprezentuj e adres URL strony

DziałkaBudowlana

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_dzi alki_PK	numer_dzi alki_PK	SmallInt	-	yes	yes	Reprezentuj e numer jednoznacz nie identyfikujac y działke (nie numer w sensie adresu!) - klucz glówny
powierzch nia	powierzch nia	Float	10	no	no	Reprezentuj e powierzchni e działki
adres_dzi alki	adres_dzi alki	AdresD	-	no	yes	Reprezentuj e adres działki - pole segmentow e
media	media	MediaD	-	no	no	Reprezentuj e media dostępne na działce (gaz, prąd,

						woda) - pole wyliczeniow e
garaz	garaz	Boolean	-	no	no	Reprezentuj e wiedze czy na dzialce znajduje sie garaz
ksiega_wi eczysta	ksiega_wi eczysta	Boolean	-	no	yes	Reprezentuj e 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_do mu_PK	numer_do mu_PK	SmallInt	-	yes	yes	Reprezentuj e numer jednoznacz nie identyfikujac y dom (nie numer domu w sensie adresu!) - klucz glowny
powierzch nia	powierzch nia	Float	2	no	no	Reprezentuj e powierzchni e domu
adres_do mu	adres_do mu	AdresD	-	no	yes	Reprezentuj e adres domu

ilosc_piet er	ilosc_piet er	SmallInt	-	no	no	Reprezentuj e ilosc pieter domu
budynki_g ospodarc ze	budynki_g ospodarc ze	VarChar	1024	no	no	Reprezentuj e wiedze na temat innych budynkow znajdujacyc h sie na dzialce z domem (garaz, komorka ogrodnicza, itd.)
ilosc_pok oi	ilosc_pok oi	SmallInt	-	no	no	Reprezentuj e ilosc pokoi (innych niz lazienka) znajdujacyc h sie w domu
ilosc_lazi enek	ilosc_lazi enek	SmallInt	-	no	no	Reprezentuj e ilosc lazienek znajdujacyc h sie w domu

Blok

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

Caption	Name	DataType	DataType param	PK	Mandatory	Description
numer_bl oku_PK	numer_bl oku_PK	SmallInt	-	yes	yes	Reprezentuj e numer jednoznacz nie identyfikujac y blok (nie

						numer bloku w sensie adresu!) - klucz glówny
ilosc_mieszkan	ilosc_mieszkan	SmallInt	-	no	no	Reprezentuje ilosc mieszkań
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 identyfikujący mieszkanie (nie numer mieszkania w sensie adresu!) - klucz glówny
ilosc_pokoi	ilosc_pokoi	SmallInt	-	no	no	Reprezentuje ilosc pokoi znajdujących się w mieszkaniu
ilosc_lazienek	ilosc_lazienek	SmallInt	-	no	no	Reprezentuje ilosc łazienek znajdujących się 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 conceptualnego - 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_zarządza na_przez	Jest_zarządza na_przez	SpoldzielniaMieszkaniowa	Zarząd	1...1 - 1...1
Posiada_stronę	Posiada stronę	SpoldzielniaMieszkaniowa	StronaInternetowa	1...1 - 0...m
Posiada_działkę	Posiada działkę	SpoldzielniaMieszkaniowa	DziałkaBudowlana	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
Zarządza_domem	Zarządza domem	AdministratorNieruchomosci	Dom	1...1 - 0...m
Zarządza_blokiem	Zarządza 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
Zarząd	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

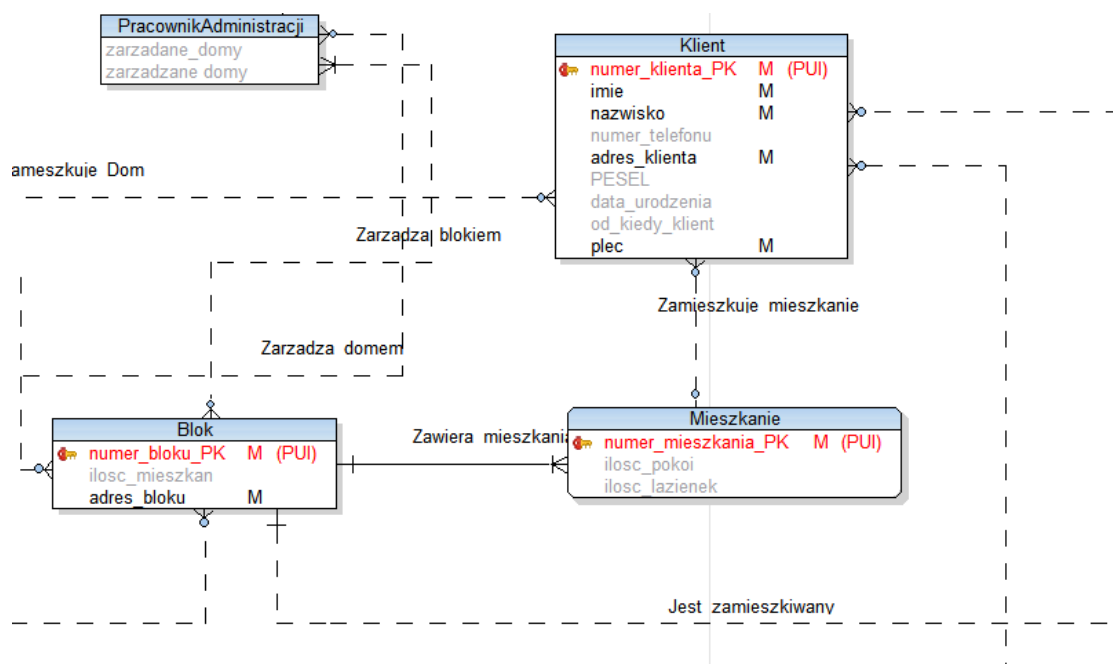
3.5. Pułapki szczelinowe i wachlarzowe

3.5.1. Wachlarzowa

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

3.5.2. Szczelinowa

W zaproponowanym modelu koncepcyjnym 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:



4. Model logiczny

4.1. Usunięcie właściwości niekompatybilnych z modelem relacyjnym

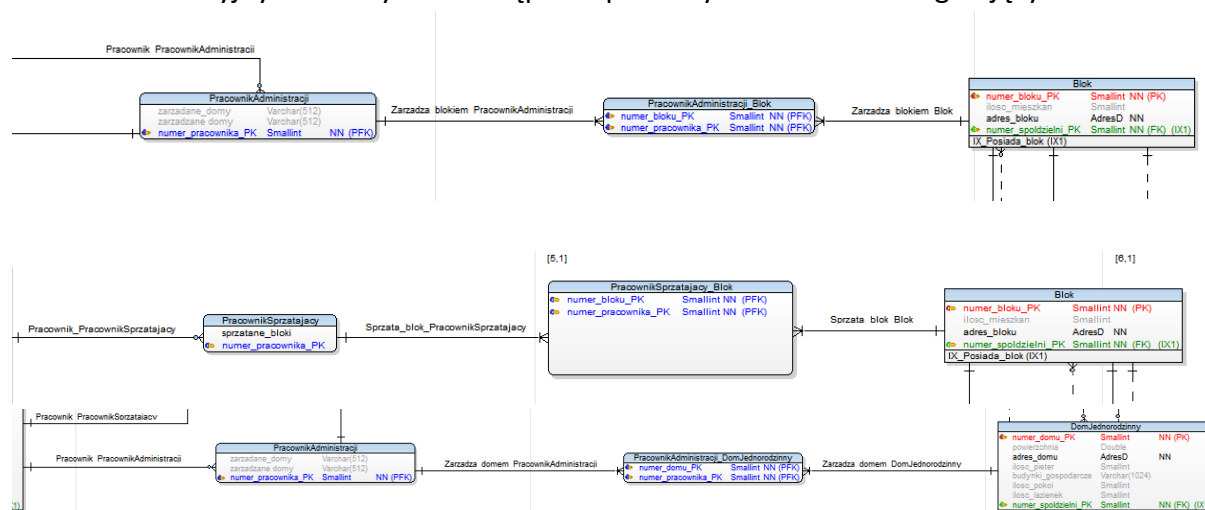
Aby przejść od zaproponowanego modelu konceptualnego do logicznego modelu relacyjnego podjęto próbę usunięcia niekompatybilności modelu konceptualnego z modelem relacyjnym poprzez następujące działania:

- 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
- Zaradza_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 tym 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 pocztowy" 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_pocztowy', a 'Kod_pocztowy' 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 "CzłonekWieZarządu".

Członkowie Zarządu:

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:

Pracownicy Księgowości:

Atrybut "dostep_do_danych_niejawnych" (finansowych) jest polem segmentowym - w tym celu stworzono osobną relację "DaneNiejawne" i linkowano ją.

Pracownicy Administracji:

Dodałem analogicznie jak dla Księgowości atrybut Poziom, a te dwa "zarządzane_domy" i "zarządzane_bloki" to chyba można wywalić, skoro i tak potem idą te bridge do bloków i domów

PracownicySprzątający:

Tutaj jak wyżej chyba też można wyrzucić ten atrybut "sprzatane_bloki" bo są też 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.

5. Model fizyczny

5.1. Projekt transakcji i weryfikacja ich wykonalności

5.2. Strojanie bazy danych poprzez dobór indeksów

5.3. Skrypt SQL generujący bazę danych

5.4. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

6. Bibliografia