

Marcin Łobacz, Aleksandr Rogaczewski

Politechnika Warszawska
Semestr 22Z
Bazy Danych i BigData

Dom Seniora

9 czerwca 2024

Spis treści

1. Zakres i cel projektu (opis założeń funkcjonalnych projektowanej bazy danych)	3
2. Definicja systemu	3
2.1. Perspektywy użytkowników	3
2.1.1. Dyrektor	3
2.1.2. Menadżer	3
2.1.3. Pracownik	3
2.1.4. Pracownik medyczny	3
2.2. Senior	3
3. Model konceptualny	4
3.1. Definicja zbiorów encji określonych w projekcie (decyzje projektowe)	4
3.2. Ustalenie związków między encjami i ich typów	7
3.3. Dodatkowe reguły integralnościowe (reguły biznesowe)	8
3.4. Klucze kandydujące i główne (decyzje projektowe)	8
3.5. Schemat ER na poziomie konceptualnym	8
3.6. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady	10
4. Model logiczny	10
4.1. Charakterystyka modelu relacyjnego	10
4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady	10
4.3. Proces normalizacji – analiza i przykłady	11
4.3.1. 1NF	11
4.3.2. 2NF	11
4.3.3. 3NF	12
4.4. Schemat ER na poziomie modelu logicznego	12
4.5. Więzy integralności	14
4.6. Proces denormalizacji – analiza i przykłady	14
5. Faza fizyczna	14
5.1. Projekt transakcji i weryfikacja ich wykonalności	14
5.2. Strojenie bazy danych – dobór indeksów	16
5.3. Skrypt SQL zakładający bazę danych	16
5.4. Przykłady zapytań i poleceń SQL uzupełniających BD	26
5.5. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych	29

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

Celem projektu jest opracowanie relacyjnej bazy danych na conceptualnym i logicznym poziomach. Zaimplementować bazę danych w sposób fizyczny. Baza danych zawiera zbiór informacji o Domie Seniora. W tym celu wykorzystaliśmy z systemu zarządzania bazami danych “Oracle”.

Podczas tego projektu skorzystaliśmy z następnego oprogramowania

- Toad Data Modeler 7.3
- SQLDeveloper
- Oracle Database

2. Definicja systemu

2.1. Perspektywy użytkowników

Baza danych będzie wykorzystywana przez następujących użytkowników:

2.1.1. Dyrektor

Dyrektor odpowiada za cały Dom. Ma dostęp do wszystkich danych i funkcji. Może zatrudniać pracowników, kupować, nowy sprzęt medyczny, organizować i sprzedawać zajęcia. Ma uprawnienia administratora w Oracle.

2.1.2. Menadżer

Menadżer-osoba, która zajmuje się zakwaterowaniem seniorów. Ma dostęp do swoich danych.

2.1.3. Pracownik

Pracownik- ma dostęp do swoich danych, to są osoba która pracuje w tym Domie

2.1.4. Pracownik medyczny

Osoby, które mają dostęp do swoich danych i danych seniorów. Ma również wgląd do pomieszczeń. Pracownik medyczny i opiekun zajmują się klientem i jego zdrowiem

2.2. Senior

Ma dostęp do pokoju mieszkalnego, gabinetu oraz sali gier. Może też sprawdzać dostępne zajęcia.

3. Model konceptualny

3.1. Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

Dom- w naszym przypadku istnieje tylko jedna encja Domu, definiuje i określa strukturę domu

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_domu	SmallInt	Obowiązkowy	Klucz, główny pozwala na identyfikację domu Identyfikator domu
Nazwa	VarChar(20)	Obowiązkowy	Nazwa domu
Adres	VarChar(400)	Obowiązkowy	Adres domu
Data_zalozenia	Date	Obowiązkowy	Data założenia domu
Wlasciciel	VarChar(600)	Obowiązkowy	Dane Własciciela
Telefon	VarChar(12)	Obowiązkowy	Telefon kontaktowy
Telefon_2	VarChar(12)	Nieobowiązkowy	Drugi telefon kontaktowy
Kod Pocztowy	VarChar(6)	Obowiązkowy	Kod pocztowy
Email	VarChar(40)	Nieobowiązkowy	Adres e-mail Domu
Strona_www	VarChar(40)	Nieobowiązkowy	Strona www domu

Pracownik - encja reprezentująca pojedynczego pracownika.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_pracownika	Integer	Obowiązkowy	Identyfikator pracownika
Imie	VarChar(20)	Obowiązkowy	Imię pracownika
Nazwisko	VarChar(30)	Obowiązkowy	Nazwisko pracownika
Data_urodzenia	Date	Obowiązkowy	Data urodzenia pracownika
PESEL	Character(11)	Nieobowiązkowy	Numer PESEL pracownika
Plec	PlecD	Obowiązkowy	Płeć pracownika
Adres	VarChar(400)	Obowiązkowy	Adres zamieszkania pracownika
Stanowisko	VarChar(50)	Obowiązkowy	Stanowisko zajmowane przez pracownika
Data_zatrudnienia	Date	Obowiązkowy	Data zatrudnienia pracownika
Data_zwolnienia	Date	Nieobowiązkowy	Data zwolnienia pracownika
Nr_telefonu	VarChar(15)	Nieobowiązkowy	Numer telefonu pracownika
Nr_konta	VarChar(20)	Nieobowiązkowy	Numer konta bankowego pracownika
Email	VarChar(30)	Nieobowiązkowy	Adres email pracownika

Pracownik_medyczny - stanowisko pracownika medycznego

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_licencji	VarChar(20)	Obowiązkowy	Numer licencji pracownika medycznego
Specjalizacja	VarChar(30)	Obowiązkowy	Specjalizacja pracownika medycznego

Zajecie - określa zajęcie oferowaną przez dom seniora.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_zajecia	Integer	Obowiązkowy	Klucz, unikatowy numer zajęcia
Nazwa	VarChar(20)	Obowiązkowy	Nazwa zajęcia
Opis	VarChar(800)	Obowiązkowy	Opis zajęcia
Rodzaj_zajecia	RodzajZajeciaD	Obowiązkowy	Rodzaj zajęcia
Cena	Money	Nieobowiązkowy	Cena zajęcia

Senior - encja reprezentująca pojedynczego Seniora.

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_Seniora	Integer	Obowiązkowy	Klucz, pozwala na identyfikację domu
Imie	VarChar(20)	Obowiązkowy	Imie Seniora
Nazwisko	VarChar(30)	Obowiązkowy	Nazwisko Seniora
Data_urodzenia	Date	Obowiązkowy	Data urodzenia Seniora
PESEL	Character(11)	Nieobowiązkowy	PESEL Seniora
Choroby	VarChar(400)	Obowiązkowy	Choroby Seniora
Plec	PlecD	Obowiązkowy	Płeć Seniora
Nr_telefonu	VarChar(15)	Nieobowiązkowy	Numer telefonu Seniora
Data_zakwaterowania	Date	Obowiązkowy	Data zakwaterowania Seniora
Data_smierci	Date	Nieobowiązkowy	Data śmierci Seniora

Pomieszczenie - określa pomieszczenie w domu seniora

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_pomieszczenia	Integer	Obowiązkowy	Numer pomieszczenia, klucz identyfikujący pomieszczenie
Nr_pietra	SmallInt	Obowiązkowy	Numer piętra na którym znajduje się pomieszczenie
Czy_wykorzystywane	Boolean	Obowiązkowy	Opisuje czy dane pomieszczenie jest już wykorzystywane
Wielkość	Decimal(4,2)	Nieobowiązkowy	Wielkość pomieszczenia w metrach kwadratowych

Gabinet - określa gabinet w domu seniora

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Rodzaj	VarChar(30)	Obowiązkowy	Rodzaj gabinetu
Sprzet.medyczny	VarChar(400)	Nieobowiązkowy	Opis sprzętu medycznego
Sprzet.rehabilitacyjny	VarChar(400)	Nieobowiązkowy	Opis sprzętu rehabilitacyjnego

Pokoj_mieszkalny - określa pokój mieszkalny w domu seniora

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Ilosc_miejsc	IloscMiejscD	Obowiązkowy	Opisuje ilu osobowy jest pokój
Standard	StandardD	Obowiązkowy	Opisuje standard pokoju

Sala_gier - określa salę gier w domu seniora

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Czy_telewizor	Boolean	Obowiązkowy	Czy mamy telewizor w pokoju
Liczba_komputerow	SmallInt	Obowiązkowy	Liczba komputerów
Liczba_gier_planszowych	SmallInt	Obowiązkowy	Liczba gier planszowych
Liczba_krzyzowek	SmallInt	Obowiązkowy	Liczba krzyżówek
Czy_Radio	Boolean	Obowiązkowy	Czy mamy radio w pokoju

3.2. Ustalenie związków między encjami i ich typów

Relacje		Nazwa relacji	Typ związku	Typ uczestnictwa		Stopień związku
Dom	Pracownik	Dom zatrudnia pracownika	1:n	Obowiązkowy	Nieobowiązkowy	Binarny
Dom	Zajecie	Dom oferuje zajecie	1:n	Obowiązkowy	Nieobowiązkowy	Binarny
Dom	Pomieszczenie	Dom posiada pomieszczenie	1:n	Obowiązkowy	Obowiązkowy	Binarny
Dom	Senior	Dom posiada seniora	1:n	Obowiązkowy	Nieobowiązkowy	Binarny
Pracownik	Senior	Pracownik opiekuje się seniorem	m:n	Obowiązkowy	Nieobowiązkowy	Binarny
Pracownik	Zajecie	Pracownik wykonuje zajecie	m:n	Nieobowiązkowy	Nieobowiązkowy	Binarny
Pracownik	Pomieszczenie	Pracownik obsługuje pomieszczenie	m:n	Obowiązkowy	Nieobowiązkowy	Binarny
Pracownik	Pracownik	Pracownik kieruje pracownikiem	1:n	Obowiązkowy	Nieobowiązkowy	Binarny
Senior	Zajecie	Senior korzysta z zajęcia	m:n	Nieobowiązkowy	Nieobowiązkowy	Binarny
Senior	Pomieszczenie	Senior korzysta z pomieszczenia	m:n	Nieobowiązkowy	Obowiązkowy	Binarny
Zajecie	Pomieszczenie	Zajecie jest prowadzone w pomieszczeniu	m:n	Obowiązkowy	Nieobowiązkowy	Binarny

Przyjęte założenia podczas tworzenia danego typu związku:

- **Dom - Pracownik:** Jeden dom zatrudnia wielu pracowników, pracownik może być zatrudniony przez jeden dom.
- **Dom - zajęcia:** Jeden dom oferuje wiele zajęć.
- **Dom - Pomieszczenie:** Jeden dom posiada przynajmniej jedno pomieszczenie. Pomieszczenie nie może istnieć bez domu.
- **Dom - Senior:** Jeden dom posiada wiele seniorów. Dom nie musi posiadać seniora. Senior musi być przypisany do domu.
- **Pracownik- Senior:** Wiele pracowników może opiekować się nad wieloma seniorami. Każdy senior powinien być pod opieką przynajmniej jednego pracownika, ale nie każdy pracownik powinien opiekować się seniorami.
- **Pracownik - Zajęcie:** Wiele pracowników może wykonywać wiele zajęć. Pracownik nie musi wykonywać zajęcia i zajecie nie musi być wykonywane przez pracownika.
- **Pracownik - Pomieszczenie:** Wiele pracowników obsługuje wiele pomieszczeń. Pracownik może nie być przypisany do żadnego z pomieszczeń. Pomieszczenie może być nie niewykorzystywane, więc może nie być przypisane do pracownika
- **Pracownik - Pracownik:** Związek rekurencyjny, pracownik kieruje pracownikiem.
- **Senior - Zajęcie:** Wiele Seniorów może korzystać z wielu zajęć. Senior może nie korzystać z żadnego zajęcia, zajecie nie jest "przypisane do Seniora".
- **Senior - Pomieszczenie:** Wiele seniorów korzysta z wielu pomieszczeń. Senior musi być przypisany do pokoju mieszkalnego. I pokój mieszkalny musi mieć przypisanego seniora. W innych specjalizacjach obowiązku takiego nie ma.
- **Zajęcie - Pomieszczenie:** Wiele zajęć może być prowadzone w pomieszczeniu. Zajęcia nie muszą być prowadzone w pomieszczeniu, pomieszczenie nie musi być przypisane do pomieszczenia.

Nowe dziedziny

Na poziomie konceptualnym określiliśmy nowe dziedziny:

- **PlecD** - dziedzina na podstawie typu Character(1), z możliwością dopuszczenia opcji 'K'-(kobieta) i 'M'-(mężczyzna). Wykorzystujemy ją w atrybutach Plec w encji Pracownik i Senior.

- **StandardID** - dziedzina utworzona na podstawie typu Character(1). Do wyboru są standard: 3, 4 lub 5 gwiazdek. Tą dziedzinę wykorzystujemy w specjalizacji Pomieszczenia - Pokoj_mieszkalny.
- **IloscMiejscD** - dziedzina utworzona na podstawie typu Character(1). Wykorzystujemy ją w specjalizacji Pomieszczenia - Pokoj_mieszkalny. Do wyboru mamy 1, 2, 3 osobowe pokoje.
- **RodzajZajęciaD** - dziedzina utworzona na podstawie typu Character(15) i może przyjmować znaczenia 'MEDYCZNE', 'REKREACYJNE', 'KULTURALNE', wykorzystujemy w encji Zajęcie

3.3. Dodatkowe reguły integralnościowe (reguły biznesowe)

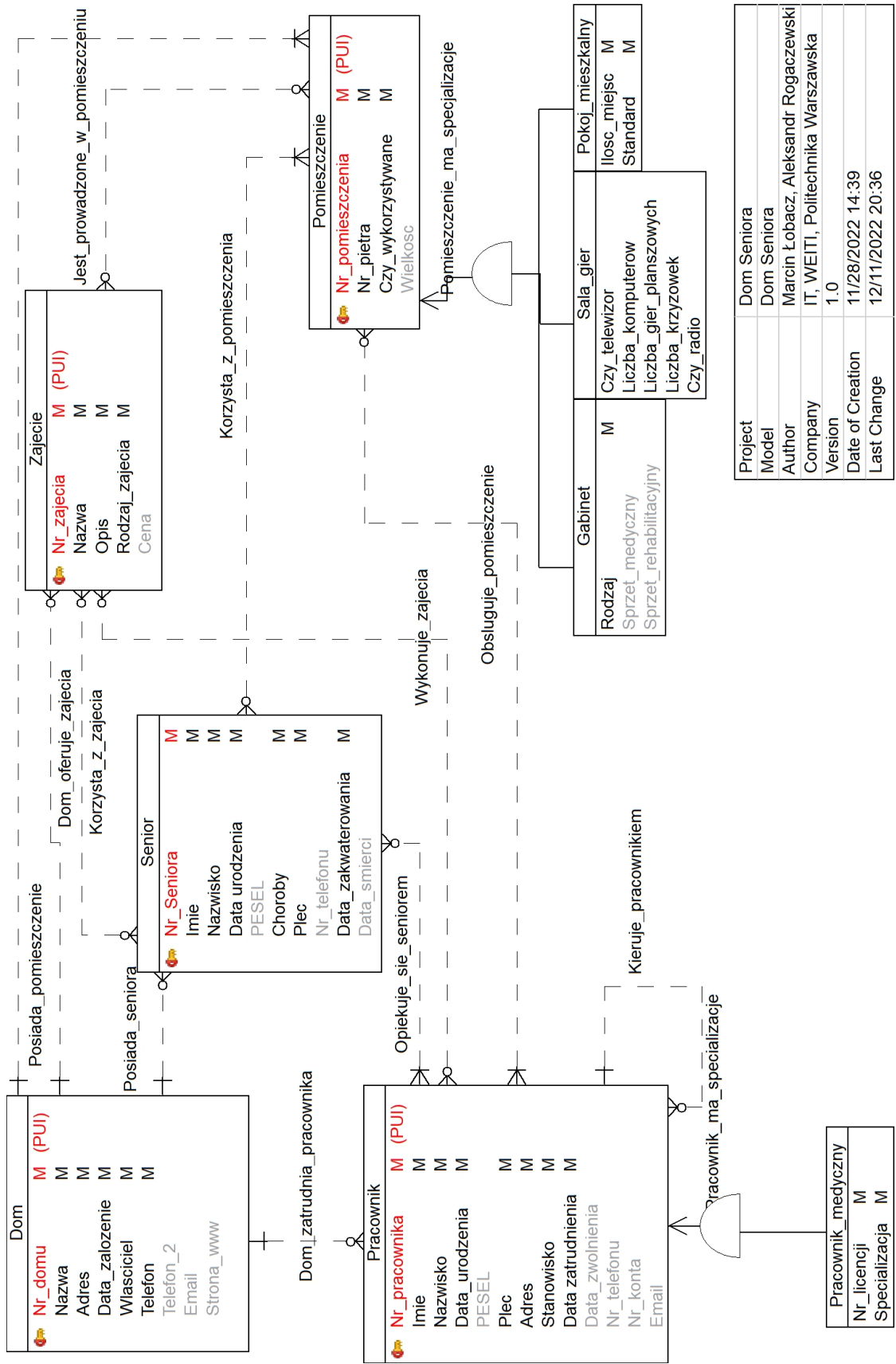
Reguły integralnościowe są ograniczeniami nakładanymi na bazy danych. Określają jakie operacje, prowadzące do modyfikacji danych, są dozwolone. W tworzeniu naszego modelu konceptualnego, pamiętaliśmy o integralności encji - tabela powinna mieć unikalny klucz główny. Zwróciliśmy również uwagę na to, aby zgadzały się zawartości pól rekordu z określonym typem rekordu.

3.4. Klucze kandydujące i główne (decyzje projektowe)

Wybór kluczy głównych spośród kluczy kandydujących nie był zbyt trudny, klucze główne encji są typowo kluczami sztucznymi, zaczynającymi się od nr, reszta to tylko klucze potencjalne. W poniższej tabeli znajdują się klucze kandydujące i główne. Klucz kandydujący typu PESEL nie jest najlepszym rozwiązaniem, ponieważ istnieją przypadki, w których osoba nie ma numeru PESEL. Inne wymienione klucze kandydujące, również nie są najbardziej optymalne. Klucze sztuczne zajmują mniej zasobów, czyniąc je najlepszym kandydatem na klucz główny.

Encja	Klucz główny	Inne klucze kandydujące
Dom	Nr_domu	(Nazwa, Adres)
Pracownik	Nr_pracownika	PESEL, (Imie, Nazwisko, Data_urodzenia, Plec)
Senior	Nr_seniora	PESEL, (Imie, Nazwisko, Data_urodzenia, Plec)
Zajęcie	Nr_zajecia	(Nazwa, Data_Od, Data_Do)
Pomieszczenie	Nr_pomieszczenia	

3.5. Schemat ER na poziomie konceptualnym



3.6. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

Pułapka szczelinowa

Pułapka mogłaby wystąpić w przypadku braku połączenia encji Pomieszczenia z encją Domy. Wtedy stracilibyśmy informacje o liczbie pomieszczeń w domu, ale nadalbyśmy mieli informacje o zajęciach prowadzonych w pomieszczeniu i o osobie, która obsługuje to pomieszczenie.

Pułapka wachlarzowa

W naszym przypadku pułapka wachlarzowa może wystąpić przy braku relacji Pracownik-Senior. Przy jej braku pracownik byłby niejednoznacznie przypisany do seniora. Dodanie relacji likwiduje ten problem. Kolejnym przypadkiem jest brak relacji między Senior-Pomieszczenie. Senior musi być przypisany do pomieszczenia. Brak relacji powoduje niejednoznaczność przypisania seniora do pomieszczenia.

W finalnym schemacie poziomu konceptualnego pułapki nie występują.

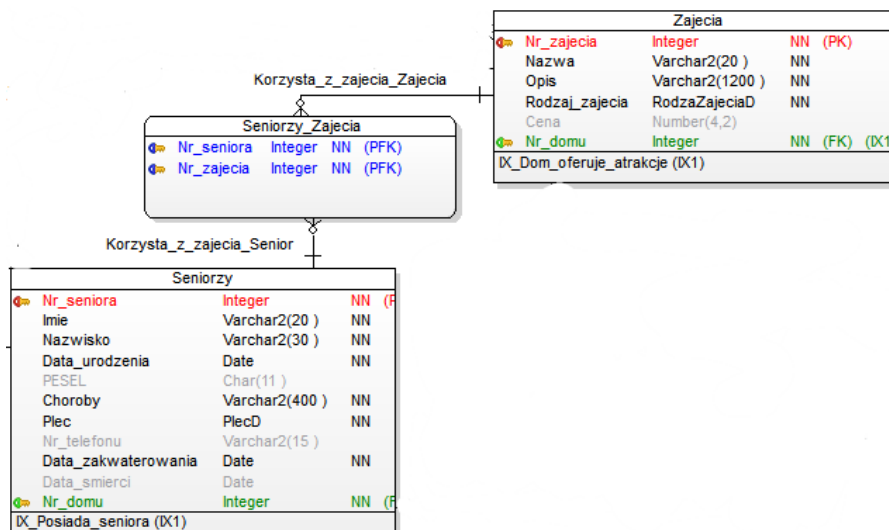
4. Model logiczny

4.1. Charakterystyka modelu relacyjnego

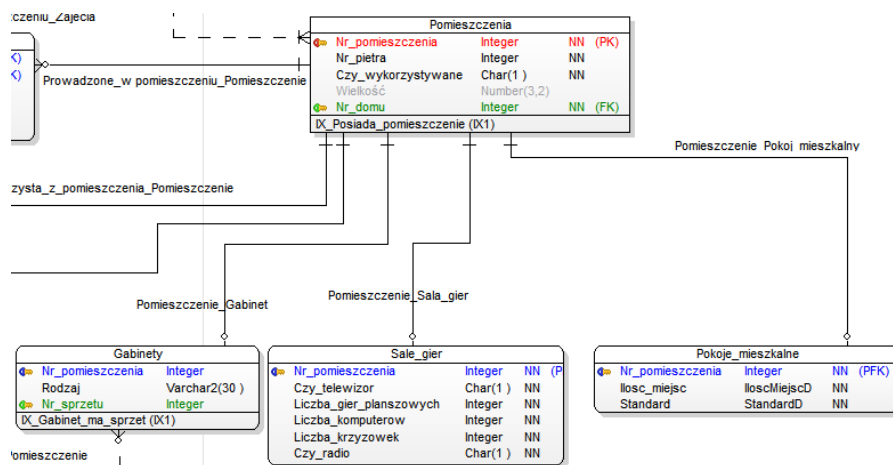
Skorzystaliśmy z funkcji automatycznego generowania modelu logicznego programu Toad Data Modeler. W procesie przekształcania do modelu logicznego związku wiele do wielu zostały zastąpione tablicami bridżującymi. Dodatkowo zwróciliśmy uwagę na to, czy automatyczny proces oddał nasze założone wymagania dostępności. Program przekształcił niektóre typy na przykład Money w Number(10,2). Wartości mandatory zamieniają się na NOT NULL. Automatyczna konwersja nie jest idealna, więc wykonaliśmy poprawki usuwające niekompatybilności z modelem relacyjnym. Dokonałiśmy także normalizacji bazy danych.

4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

Nazwy encji: Zamieniliśmy nazwy encji z liczby pojedynczej na mnogą. W modelu konceptualnym encje reprezentują pojedyncze obiekty, w modelu logicznym reprezentują wiele krótek.



Specializacje: W modelu logicznym specjalizacje zostały zastąpione relacjami.



4.3. Proces normalizacji – analiza i przykłady

4.3.1. 1NF

Zgodnie z definicją pierwszej postaci normalnej wartości atrybutów powinny być atomowe, czyli wszystkie atrybuty powinny być nierozkładalne. Analizując naszą bazę danych, zauważamy, że Adres i Właściciel w encji Dom i sprzęt w encji Gabinety, to atrybuty rozkładalne. Więc tworzymy sobie 3 nowe encje odpowiednio Adres, Właściciel i Sprzet

Adres

Nr_adresu	Integer	Obowiązkowy
Miasto	VarChar(30)	Obowiązkowy
Ulica	VarChar(30)	Obowiązkowy
Nr_lokalu	VarChar(4)	Obowiązkowy

Właściciel

Nr_wlasciciela	Integer	Obowiązkowy
Imie	VarChar(20)	Obowiązkowy
Nazwisko	VarChar(30)	Obowiązkowy

Sprzet

Nr_Sprzetu	Integer	Obowiązkowy
Typ	VarChar(30)	Obowiązkowy
Model	VarChar(15)	Obowiązkowy
Marka	VarChar(30)	Obowiązkowy
Opis	VarChar(400)	Obowiązkowy

4.3.2. 2NF

Z definicji, mamy drugą postać normalną wtedy, gdy klucze główne wszystkich relacji nie są złożone, a składają się z jednego atrybutu i jeżeli każdy atrybut nie wchodzący w skład klucza zależy od klucza, a nie od jego części. W naszym przypadku po przekształceniu do 1NF, Baza Danych już jest w drugiej postaci normalnej

4.3.3. 3NF

Z definicji mamy trzecią postać normalną wtedy, gdy każdy atrybut nie wchodzący w skład klucza zależy od całego klucza, a nie od jego części. Nasza baza jest w 3 Postaci normalnej, jedyną rzeczą którą tu zrobiliśmy jest wyniesienie atrybutów pola sprzęty (model i marka) do oddzielnych encji i powiązanie ich odpowiednimi związkami "sprzet ma model", "model ma markę"

4.4. Schemat ER na poziomie modelu logicznego

4.5. Więzy integralności

Celem wyznaczenia więzów integralności określono wymaganą obecność danych, więzy dziedzin atrybutów, integralność encji, integralność referencyjna i więzy ogólne. Więc zadaliśmy o to żeby, wartości kluczy głównych były unikatowymi (można to zrobić przy pomocy opcji UNIQUE albo PRIMARY KEY, różnica w tym, że PRIMARY KEY nie może przyjmować znaczenia NULL), wszystkie pola powinny być nierozkładalne (atomowe), co zrobiliśmy przy normalizacji do pierwszej postaci. Najważniejsze dane, których będziemy potrzebowali do prawidłowego działania bazy danych nie mogą przyjmować wartości NULL.

4.6. Proces denormalizacji – analiza i przykłady

Stosowanie denormalizacji skutkuje redundancją danych oraz utratą spójności danych, jednakże zyskujemy większą szybkość wyszukiwania danych. Po sprowadzeniu bazy danych do trzeciej postaci normalnej, stwierdziliśmy, że nie potrzebujemy denormalizacji. Na etapie projektowania chcieliśmy zamienić w encji *Sala_gier* atrybut *Liczba_komputerow*, *Liczba_telewizorow*, *Czy_radio* na encje *Urządzenia_elektroniczne*, ale po przemyśleniu naszego modelu biznesowego, zdecydowaliśmy się dokonać procesu denormalizacji i zamiast tej encji, użyć atrybutów "*Liczba_komputerow*", *Liczba_telewizorow*, *Czy_radio*, *Czy_telewizor*. Dotyczyło to także atrybutu *Liczba_krzyzowek*. Stwierdziliśmy, że szybkość wyszukiwania danych jest dla nas najważniejsza dlatego nie stworzyliśmy encji *Urządzenia_elektroniczne*.

5. Faza fizyczna

5.1. Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Potrzebne Dane	Weryfikacja wykonalności
Podgląd informacji o domu seniora	Domy, Adresy, Poczty	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o domu seniora	Domy, Adresy, Poczty	Wykonalne
Podgląd informacji o zajęciach	Domy, Zajęcia, Pomieszczenia	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o zajęciach	Domy, Zajęcia, Pomieszczenia	Wykonalne
Podgląd informacji o Seniorach	Dom, Senior	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o seniorach	Domy, Seniorzy	Wykonalne
Podgląd informacji pracowników	Pracownicy, Domy, Adresy	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o pracownikach	Pracownicy, Domy, Adresy	Wykonalne
Podgląd informacji o pracownikach medycznych	Pracownicy, Pracownicy medyczni	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o pracownikach medycznych	Pracownicy, Pracownicy medyczni	Wykonalne
Podgląd informacji o stanowiskach	Pracownicy, Stanowiska	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o stanowiskach	Pracownicy, Stanowiska	Wykonalne
Podgląd informacji o właścicielach	Domy, Właściciele, Adresy	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o właścicielach	Domy, Właściciele, Adresy	Wykonalne
Podgląd informacji o pomieszczeniach	Domy, Pomieszczenia	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o pomieszczeniach	Domy, Pomieszczenia	Wykonalne
Podgląd informacji w gabinetach	Pomieszczenia, Gabinety	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o gabinetach	Pomieszczenia, Gabinety	Wykonalne
Podgląd informacji o salach gier	Pomieszczenia, Sale gier	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o salach gier	Pomieszczenia, Sale gier	Wykonalne
Podgląd informacji o pokojach mieszkalnych	Pomieszczenia, Pokoje Mieszkalne	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o pokojach mieszkalnych	Pomieszczenia, Pokoje Mieszkalne	Wykonalne
Podgląd informacji o zajęciach wykorzystywanych przez seniorów	Seniorzy, Zajęcia, Pomieszczenia	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o zajęciach wykorzystywanych przez seniorów	Seniorzy, Zajęcia, Pomieszczenia	Wykonalne
Podgląd informacji o zajęciach prowadzonych przez pracowników	Pracownicy, Zajęcia, Pomieszczenia	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o zajęciach prowadzonych przez pracowników	Pracownicy, Zajęcia, Pomieszczenia	Wykonalne
Podgląd informacji o wykorzystywaniu pomieszczenia przez pracownika	Pracownicy, Pomieszczenia	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o wykorzystywaniu pomieszczenia przez pracownika	Pracownicy, Pomieszczenia	Wykonalne
Podgląd informacji o pomieszczeniu wykorzystywanym przez seniorów	Seniorzy, Pomieszczenia	Wykonalne
Dodawanie/modyfikowanie/usuwanie informacji o pomieszczeniu wykorzystywanym przez seniorów	Seniorzy, pomieszczenia	Wykonalne

5.2. Strojzenie bazy danych – dobór indeksów

Domy

```
CREATE INDEX IX_Dom_ma_adres ON Domy (Nr_adresu)/
```

Pracownicy

```
CREATE INDEX IX_Kieruje_pracownikiem ON Pracownicy (Nr_kierownika)/
```

```
CREATE INDEX IX_Dom_zatrudnia_pracownika ON Pracownicy (Nr_domu)/
```

```
CREATE INDEX IX_Pracownik_Posiada_Stanowisko ON Pracownicy (Nr_Stanowiska)/
```

```
CREATE INDEX IX_Pracownik_ma_adres ON Pracownicy (Nr_adresu)/
```

Wlasciciele

```
CREATE INDEX IX_Dom_ma_wlasciciela ON Wlasciciele (Nr_domu)/
```

```
CREATE INDEX IX_Wlasciciel_ma_adres ON Wlasciciele (Nr_adresu)/
```

Pomieszczenia

```
CREATE INDEX IX_Posiada_pomieszczenie ON Pomieszczenia (Nr_domu)/
```

Gabinety

```
CREATE INDEX IX_Gabinet_ma_sprzet ON Gabinety (Nr_sprzetu)/
```

Zajecia

```
CREATE INDEX IX_Dom_oferuje_atrakcje ON Zajecia (Nr_domu)/
```

Seniorzy

```
CREATE INDEX IX_Posiada_seniora ON Seniorzy (Nr_domu)/
```

Adresy

```
CREATE INDEX IX_Adres_ma_poczte ON Adresy (Nr_poczty)/
```

Sprzety

```
CREATE INDEX IX_Sprzet_ma_model ON Sprzety (Nr_model)/
```

Modele

```
CREATE INDEX IX_Marka_ma_model ON Modele (Nr_marki)/
```

5.3. Skrypt SQL zakładający bazę danych

```
/*
```

```
Created: 07.12.2022
```

```
Modified: 11.12.2022
```

```
Project: Dom Seniora
```

```
Model: Dom Seniora
```

```
Company: WEITI, Politechnika Warszawska
```

```
Author: Marcin Łobacz, Aleksandr Rogaczewski
```

```
Version: 1.0
```

```
Database: Oracle 19c
```

```
*/
```

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

```
-- Table Sale_gier
```

```
CREATE TABLE Sale_gier(
```



```

    Nr_pomieszczenia Integer NOT NULL,
    Czy_telewizor Char(1 ) NOT NULL,
    Liczba_gier_planszowych Integer NOT NULL,
    Liczba_komputerow Integer NOT NULL,
    Liczba_krzyzowek Integer NOT NULL,
    Czy_radio Char(1 ) NOT NULL
)
/

-- Add keys for table Sale_gier

ALTER TABLE Sale_gier ADD CONSTRAINT Unique_Identifier6
PRIMARY KEY (Nr_pomieszczenia)
/

-- Table Gabinety

CREATE TABLE Gabinety(
    Nr_pomieszczenia Integer NOT NULL,
    Rodzaj Varchar2(30 ) NOT NULL,
    Nr_sprzetu Integer NOT NULL
)
/

-- Create indexes for table Gabinety

CREATE INDEX IX_Gabinet_ma_sprzet ON Gabinety (Nr_sprzetu)
/

-- Add keys for table Gabinety

ALTER TABLE Gabinety ADD CONSTRAINT GabinetPK PRIMARY KEY (Nr_pomieszczenia)
/

-- Table Zajecia

CREATE TABLE Zajecia(
    Nr_zajecia Integer NOT NULL,
    Nazwa Varchar2(20 ) NOT NULL,
    Opis Varchar2(1200 ) NOT NULL,
    Rodzaj_zajecia Varchar2(15 ) NOT NULL
        CHECK (Rodzaj_zajecia IN ('MEDYCZNE','REKREACYJNE','KULTURALNE')),
    Cena Number(4,2),
    Nr_domu Integer NOT NULL
)
/

-- Create indexes for table Zajecia

CREATE INDEX IX_Dom_oferuje_zajecie ON Zajecia (Nr_domu)
/

-- Add keys for table Zajecia

ALTER TABLE Zajecia ADD CONSTRAINT ZajeciaPK
PRIMARY KEY (Nr_zajecia)

```

```

/

-- Table Seniorzy

CREATE TABLE Seniorzy(
    Nr_seniora Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Data_urodzenia Date NOT NULL,
    PESEL Char(11 ),
    Choroby Varchar2(400 ) NOT NULL,
    Plec Char(1 CHAR) NOT NULL
        CHECK (Plec IN ('K','M')),
    Nr_telefonu Varchar2(15 ),
    Data_zakwaterowania Date NOT NULL,
    Data_smierci Date,
    Nr_domu Integer NOT NULL
)
/

-- Create indexes for table Seniorzy

CREATE INDEX IX_Posiada_seniora ON Seniorzy (Nr_domu)
/

-- Add keys for table Seniorzy

ALTER TABLE Seniorzy ADD CONSTRAINT SeniorPK
PRIMARY KEY (Nr_seniora)
/

-- Table Domy

CREATE TABLE Domy(
    Nr_domu Integer NOT NULL,
    Nazwa Varchar2(20 ) NOT NULL,
    Data_zalozenia Date NOT NULL,
    Email Varchar2(40 ),
    Telefon Varchar2(15 ) NOT NULL,
    Telefon_2 Varchar2(15 ),
    Strona_www Varchar2(40 ),
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Domy

CREATE INDEX IX_Dom_ma_adres ON Domy (Nr_adresu)
/

-- Add keys for table Domy

ALTER TABLE Domy ADD CONSTRAINT DomPK
PRIMARY KEY (Nr_domu)
/

```

```

-- Table Pracownicy_medyczni

CREATE TABLE Pracownicy_medyczni(
    Nr_pracownika Integer NOT NULL,
    Nr_licencji Varchar2(20 ) NOT NULL,
    Specjalizacja Varchar2(30 ) NOT NULL
)
/

-- Add keys for table Pracownicy_medyczni

ALTER TABLE Pracownicy_medyczni ADD CONSTRAINT
Unique_Identifier4 PRIMARY KEY (Nr_pracownika)
/

-- Table Pracownicy

CREATE TABLE Pracownicy(
    Nr_pracownika Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Data_urodzenia Date NOT NULL,
    PESEL Char(11 ),
    Plec Char(1 CHAR) NOT NULL
        CHECK (Plec IN ('K','M')),
    Data_zatrudnienia Date NOT NULL,
    Data_zwolnienia Date,
    Nr_telefonu Varchar2(15 ),
    Nr_konta Varchar2(20 ),
    Email Varchar2(30 ),
    Nr_kierownika Integer NOT NULL,
    Nr_domu Integer NOT NULL,
    Nr_Stanowiska Integer NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Pracownicy

CREATE INDEX IX_Kieruje_pracownikiem ON Pracownicy (Nr_kierownika)
/

CREATE INDEX IX_Dom_zatrudnia_pracownika ON Pracownicy (Nr_domu)
/

CREATE INDEX IX_Pracownik_Posiada_Stanowisko ON Pracownicy (Nr_Stanowiska)
/

CREATE INDEX IX_Pracownik_ma_adres ON Pracownicy (Nr_adresu)
/

-- Add keys for table Pracownicy

ALTER TABLE Pracownicy ADD CONSTRAINT
PracownikPK PRIMARY KEY (Nr_pracownika)
/

```

```

-- Table Pomieszczenia

CREATE TABLE Pomieszczenia(
    Nr_pomieszczenia Integer NOT NULL,
    Nr_pietra Integer NOT NULL,
    Czy_wykorzystywane Char(1 ) NOT NULL,
    Wielkość Number(4,2),
    Nr_domu Integer NOT NULL
)
/

-- Create indexes for table Pomieszczenia

CREATE INDEX IX_Posiada_pomieszczenie ON Pomieszczenia (Nr_domu)
/

-- Add keys for table Pomieszczenia

ALTER TABLE Pomieszczenia ADD CONSTRAINT
PomieszczeniePK PRIMARY KEY (Nr_pomieszczenia)
/

-- Table Pokoje_mieszkalne

CREATE TABLE Pokoje_mieszkalne(
    Nr_pomieszczenia Integer NOT NULL,
    Ilosc_miejsc Char(1 CHAR) NOT NULL
        CHECK (Ilosc_miejsc IN ('1', '2', '3')),
    Standard Char(1 CHAR) NOT NULL
        CHECK (Standard IN ('3','4','5'))
)
/

-- Add keys for table Pokoje_mieszkalne

ALTER TABLE Pokoje_mieszkalne ADD CONSTRAINT
Unique_Identifier7 PRIMARY KEY (Nr_pomieszczenia)
/

-- Table Seniorzy_Pomieszczenia

CREATE TABLE Seniorzy_Pomieszczenia(
    Nr_seniora Integer NOT NULL,
    Nr_pomieszczenia Integer NOT NULL
)
/

-- Table Pracownicy_Pomieszczenia

CREATE TABLE Pracownicy_Pomieszczenia(
    Nr_pracownika Integer NOT NULL,
    Nr_pomieszczenia Integer NOT NULL
)
/

```

```

-- Table Pracownicy_Zajecia

CREATE TABLE Pracownicy_Zajecia(
    Nr_pracownika Integer NOT NULL,
    Nr_zajecia Integer NOT NULL
)
/

-- Table Seniorzy_Zajecia

CREATE TABLE Seniorzy_Zajecia(
    Nr_seniora Integer NOT NULL,
    Nr_zajecia Integer NOT NULL
)
/

-- Table Zajecia_Pomieszczenia

CREATE TABLE Zajecia_Pomieszczenia(
    Nr_zajecia Integer NOT NULL,
    Nr_pomieszczenia Integer NOT NULL
)
/

-- Table Adresy

CREATE TABLE Adresy(
    Nr_adresu Integer NOT NULL,
    Miasto Varchar2(30 ) NOT NULL,
    Ulica Varchar2(30 ) NOT NULL,
    Nr_lokalu Varchar2(4 ) NOT NULL,
    Nr_poczty Integer NOT NULL
)
/

-- Create indexes for table Adresy

CREATE INDEX IX_Adres_ma_poczte 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'
/
COMMENT ON COLUMN Adresy.Miasto IS 'Miasto'
/
COMMENT ON COLUMN Adresy.Ulica IS 'Ulica'
/
COMMENT ON COLUMN Adresy.Nr_lokalu IS 'Numer lokalu'
/

```

```

-- Table Wlasciciele

CREATE TABLE Wlasciciele(
    Nr_wlascicela Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Nr_domu Integer NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Wlasciciele

CREATE INDEX IX_Dom_ma_wlasciciela ON Wlasciciele (Nr_domu)
/

CREATE INDEX IX_Wlasciciel_ma_adres ON Wlasciciele (Nr_adresu)
/

-- Add keys for table Wlasciciele

ALTER TABLE Wlasciciele ADD CONSTRAINT
PK_Wlasciciele PRIMARY KEY (Nr_wlascicela)
/

-- Table and Columns comments section

COMMENT ON COLUMN Wlasciciele.Nr_wlascicela IS 'Unikatowy identyfikator właściciela'
/
COMMENT ON COLUMN Wlasciciele.Imie IS 'Imie właściciela'
/
COMMENT ON COLUMN Wlasciciele.Nazwisko IS 'Nazwisko właściciela'
/

-- Table Sprzety

CREATE TABLE Sprzety(
    Nr_sprzetu Integer NOT NULL,
    Typ Varchar2(30 ) NOT NULL,
    Opis Varchar2(400 ),
    Nr_model Integer NOT NULL
)
/

-- Create indexes for table Sprzety

CREATE INDEX IX_Sprzet_ma_model ON Sprzety (Nr_model)
/

-- Add keys for table Sprzety

ALTER TABLE Sprzety ADD CONSTRAINT PK_Sprzety PRIMARY KEY (Nr_sprzetu)
/

```

```

-- Table Stanowiska

CREATE TABLE Stanowiska(
    Nr_Stanowiska Integer NOT NULL,
    Nazwa_stanowiska Varchar2(20 ) NOT NULL,
    Opis Varchar2(400 ) NOT NULL
)
/

-- Add keys for table Stanowiska

ALTER TABLE Stanowiska ADD CONSTRAINT PK_Stanowiska PRIMARY KEY (Nr_Stanowiska)
/

ALTER TABLE Stanowiska ADD CONSTRAINT Nazwa_stanowisko UNIQUE (Nazwa_stanowiska)
/

-- Table and Columns comments section

COMMENT ON COLUMN Stanowiska.Nr_Stanowiska IS 'Identyfikator stanowiska'
/
COMMENT ON COLUMN Stanowiska.Opis IS 'Opis stanowiska'
/

-- Table Poczty

CREATE TABLE Poczty(
    Nr_poczty Integer NOT NULL,
    Kod_pocztowy Char(6 ) NOT NULL,
    Poczta Varchar2(30 ) NOT NULL
)
/

-- Add keys for table Poczty

ALTER TABLE Poczty ADD CONSTRAINT PK_Poczty PRIMARY KEY (Nr_poczty)
/

ALTER TABLE Poczty ADD CONSTRAINT Kod_pocztowy UNIQUE (Kod_pocztowy)
/

-- Table Modele

CREATE TABLE Modele(
    Nr_model Integer NOT NULL,
    Kod_modelu Varchar2(15 ) NOT NULL,
    Nr_marki Integer NOT NULL
)
/

-- Create indexes for table Modele

CREATE INDEX IX_Marka_ma_model ON Modele (Nr_marki)
/

-- Add keys for table Modele

```

```

ALTER TABLE Modele ADD CONSTRAINT PK_Modele PRIMARY KEY (Nr_model)
/

-- Table and Columns comments section

COMMENT ON COLUMN Modele.Nr_model IS 'Unikatowy identyfikator modelu'
/

-- Table Marki

CREATE TABLE Marki(
    Nr_marki Integer NOT NULL,
    Nazwa_marki Varchar2(30 ) NOT NULL
)
/

-- Add keys for table Marki

ALTER TABLE Marki ADD CONSTRAINT PK_Marki PRIMARY KEY (Nr_marki)
/

-- Table and Columns comments section

COMMENT ON COLUMN Marki.Nr_marki IS 'Unikatowy identyfikator marki'
/

-- Table Pracownicy_Seniorzy

CREATE TABLE Pracownicy_Seniorzy(
    Nr_pracownika Integer NOT NULL,
    Nr_seniora Integer NOT NULL
)
/

-- Add keys for table Pracownicy_Seniorzy

ALTER TABLE Pracownicy_Seniorzy ADD CONSTRAINT PK_Pracownicy_Seniorzy
PRIMARY KEY (Nr_pracownika,Nr_seniora)
/

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

ALTER TABLE Pracownicy ADD CONSTRAINT Kieruje_pracownikiem
FOREIGN KEY (Nr_kierownika) REFERENCES Pracownicy (Nr_pracownika)
/

ALTER TABLE Seniorzy ADD CONSTRAINT Posiada_seniora
FOREIGN KEY (Nr_domu) REFERENCES Domy (Nr_domu)
/

```



```
ALTER TABLE Pomieszczenia ADD CONSTRAINT Posiada_pomieszczenie  
FOREIGN KEY (Nr_domu) REFERENCES Domy (Nr_domu)  
/
```

```
ALTER TABLE Zajecia ADD CONSTRAINT Dom_oferuje_zajecie  
FOREIGN KEY (Nr_domu) REFERENCES Domy (Nr_domu)  
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Dom_zatrudnia_pracownika  
FOREIGN KEY (Nr_domu) REFERENCES Domy (Nr_domu)  
/
```

```
ALTER TABLE Gabinety ADD CONSTRAINT Gabinet_ma_sprzet  
FOREIGN KEY (Nr_sprzetu) REFERENCES Sprzety (Nr_sprzetu)  
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_Posiada_Stanowisko  
FOREIGN KEY (Nr_Stanowiska) REFERENCES Stanowiska (Nr_Stanowiska)  
/
```

```
ALTER TABLE Wlasciciele ADD CONSTRAINT Dom_ma_wlasciciela  
FOREIGN KEY (Nr_domu) REFERENCES Domy (Nr_domu)  
/
```

```
ALTER TABLE Wlasciciele ADD CONSTRAINT Wlasciciel_ma_adres  
FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)  
/
```

```
ALTER TABLE Domy ADD CONSTRAINT Dom_ma_adres  
FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)  
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_ma_adres  
FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)  
/
```

```
ALTER TABLE Adresy ADD CONSTRAINT Adres_ma_poczte  
FOREIGN KEY (Nr_poczty) REFERENCES Poczty (Nr_poczty)
```

/

```
ALTER TABLE Modele ADD CONSTRAINT Marka_ma_model  
FOREIGN KEY (Nr_marki) REFERENCES Marki (Nr_marki)
```

/

```
ALTER TABLE Sprzety ADD CONSTRAINT Sprzet_ma_model  
FOREIGN KEY (Nr_model) REFERENCES Modele (Nr_model)
```

/

```
ALTER TABLE Pracownicy_Seniorzy ADD CONSTRAINT Opiekuje_sie_seniorem_Pracownik  
FOREIGN KEY (Nr_pracownika) REFERENCES Pracownicy (Nr_pracownika)
```

/

```
ALTER TABLE Pracownicy_Seniorzy ADD CONSTRAINT Opiekuje_sie_seniorem_Senior  
FOREIGN KEY (Nr_seniora) REFERENCES Seniorzy (Nr_seniora)
```

/

5.4. Przykłady zapytań i poleceń SQL uzupełniającej BD

```
INSERT INTO Poczty VALUES(1,'232323', 'WarPoczta');  
INSERT INTO Adresy VALUES(1,'Warszawa', 'Polna','10',1);  
INSERT INTO Adresy VALUES(2,'Warszawa','Wiejska','15',1); --adres pracownika  
INSERT INTO Adresy VALUES(3,'Warszawa','Cybernetyki','105',1); --adres wlasciciela
```

```
INSERT INTO Domy VALUES(1,'Dom Seniora',  
TO_DATE('2002-02-08','YYYY-MM-DD'), NULL, '999996997', NULL, NULL,1);
```

```
INSERT INTO Wlasciciele  
VALUES(1,'Marek','Kowalski',1,3);
```

```
INSERT INTO Stanowiska  
VALUES(1,'Rehabilitnt','Osoba odpowiedzialna za rehabilitacje');
```

```
INSERT INTO Stanowiska  
VALUES(2,'Kierownik','Zajmuje sie kierowaniem pracownikami.');
```

```
INSERT INTO Stanowiska  
VALUES(3,'Opiekun','Opiekuje się seniorami.');
```

```
INSERT INTO Pracownicy
```

```

VALUES(1,'Mirek','Dąbkowski',TO_DATE('2000-02-08','YYYY-MM-DD'),
NULL, 'M', TO_DATE('2020-02-08','YYYY-MM-DD'),
NULL, NULL, NULL, NULL,1,1,2,2);

INSERT INTO Pracownicy
VALUES(2,'Tomasz','Nowak',TO_DATE('1999-02-08','YYYY-MM-DD'),
NULL, 'M', TO_DATE('2020-04-08','YYYY-MM-DD'),
NULL, NULL, NULL, NULL,1,1,1,2);

INSERT INTO Pracownicy
VALUES(3,'Anna','Nowak',TO_DATE('1999-10-08','YYYY-MM-DD'),
NULL, 'M', TO_DATE('2021-04-08','YYYY-MM-DD'),
NULL, NULL, NULL, NULL,1,1,3,2);

INSERT INTO Pracownicy_medyczni VALUES(3,'9382928290298','Pielęgniarstwo');
INSERT INTO Pracownicy_medyczni VALUES(2,'3232323232323','Rehabilitacja');

INSERT INTO Seniorzy
VALUES(1,'Jan','Kowal',TO_DATE('1960-02-08','YYYY-MM-DD'),
NULL,'Reumatyzm, Inkontynencja', 'M', NULL,
TO_DATE('2021-02-08','YYYY-MM-DD'),NULL,1);

INSERT INTO Pracownicy_Seniorzy VALUES(3,1);

INSERT INTO Seniorzy
VALUES(2,'Anna','Kowalczyk',TO_DATE('1958-02-08','YYYY-MM-DD'),
NULL,'Reumatyzm', 'K', NULL,
TO_DATE('2022-02-08','YYYY-MM-DD'),NULL,1);

INSERT INTO Pracownicy_Seniorzy VALUES(3,2);

INSERT INTO Zajecia
VALUES(1,'Masaż','Relaksujący masaż','MEDYCZNE', 20.00, 1);
INSERT INTO Zajecia
VALUES(2,'Rehabilitacja','Zajęcia mające na celu poprawę sprawności fizycznej',
'MEDYCZNE', 0, 1);

INSERT INTO Zajecia VALUES(3,'Filozofia','Rozmowa o sensu życia','KULTURALNE', 5.02, 1);

INSERT INTO Zajecia VALUES(4,'FiZo','Rozmowa o komputerach','KULTURALNE', 5.02, 1);

INSERT INTO Zajecia VALUES(5,'PiWo','Jakie PiWo jest najlepsze','KULTURALNE', 5.02, 1);

INSERT INTO Pracownicy_Zajecia VALUES(2,1);
INSERT INTO Pracownicy_Zajecia VALUES(2,2);

INSERT INTO Seniorzy_Zajecia VALUES(1,1);
INSERT INTO Seniorzy_Zajecia VALUES(2,2);

INSERT INTO Pomieszczenia VALUES (1,1,'T',65.4,1);
INSERT INTO Pomieszczenia VALUES (2,1,'T',58.2,1);
INSERT INTO Pomieszczenia VALUES (3,1,'T',74.6,1);
INSERT INTO Pomieszczenia VALUES (4,1,'T',47.1,1);
INSERT INTO Pomieszczenia VALUES (5,1,'T',50.2,1);

```

```

INSERT INTO Pomieszczenia VALUES (6,1,'T',46.7,1);
INSERT INTO Pomieszczenia VALUES (7,2,'T',35.7,1);
INSERT INTO Pomieszczenia VALUES (8,2,'T',36.7,1);
INSERT INTO Pomieszczenia VALUES (9,2,'N',42.1,1);
INSERT INTO Pomieszczenia VALUES (10,2,'T',38.1,1);
INSERT INTO Pomieszczenia VALUES (11,2,'T',29.8,1);
INSERT INTO Pomieszczenia VALUES (12,3,'T',35.4,1);
INSERT INTO Pomieszczenia VALUES (13,3,'T',39.5,1);
INSERT INTO Pomieszczenia VALUES (14,3,'N',40.1,1);
INSERT INTO Pomieszczenia VALUES (15,3,'T',36.7,1);
INSERT INTO Pomieszczenia VALUES (16,3,'N',34.6,1);
INSERT INTO Pomieszczenia VALUES (17,3,'N',34.6,1);

INSERT INTO marki VALUES (1,'Samsung');
INSERT INTO marki VALUES (2,'Bayer');
INSERT INTO marki VALUES (3,'Medcom');

INSERT INTO modele VALUES (1,'a48947',1);
INSERT INTO modele VALUES (2,'5896df',2);
INSERT INTO modele VALUES (3,'g589g2',3);
INSERT INTO modele VALUES (4,'bn57e',1);

INSERT INTO sprzety VALUES (1,'REHABILITACYJNY',NULL,1);

INSERT INTO sprzety VALUES (2,'MEDYCZNY','nadcisnieniemetr',2);

INSERT INTO sprzety
VALUES (3,'MEDYCZNY','EKG jest kupiony przez syna Pani Asi',3);

INSERT INTO sprzety VALUES (4,'REHABILITACYJNY','Rower',4);

INSERT INTO gabinety VALUES(2,'REHABILITACYJNY',2);
INSERT INTO gabinety VALUES(3,'MEDYCZNY',3);
INSERT INTO gabinety VALUES(4,'MEDYCZNY',4);

INSERT INTO sale_gier VALUES(5,'T',15,3,47,'N');

INSERT INTO pokoje_mieszkalne VALUES(6,'2','3');
INSERT INTO pokoje_mieszkalne VALUES(7,'2','4');
INSERT INTO pokoje_mieszkalne VALUES(8,'1','5');
INSERT INTO pokoje_mieszkalne VALUES(9,'3','3');
INSERT INTO pokoje_mieszkalne VALUES(10,'3','3');
INSERT INTO pokoje_mieszkalne VALUES(11,'2','5');
INSERT INTO pokoje_mieszkalne VALUES(12,'2','3');
INSERT INTO pokoje_mieszkalne VALUES(13,'3','4');
INSERT INTO pokoje_mieszkalne VALUES(14,'3','3');
INSERT INTO pokoje_mieszkalne VALUES(15,'1','3');
INSERT INTO pokoje_mieszkalne VALUES(16,'1','4');
INSERT INTO pokoje_mieszkalne VALUES(17,'1','5');

```

```

INSERT INTO zajecia_pomieszczenia VALUES(3,5);
INSERT INTO zajecia_pomieszczenia VALUES(4,5);
INSERT INTO zajecia_pomieszczenia VALUES(5,5);
INSERT INTO zajecia_pomieszczenia VALUES(1,2);
INSERT INTO zajecia_pomieszczenia VALUES(1,2);

```

```

INSERT INTO seniorzy_pomieszczenia VALUES(1,10);
INSERT INTO seniorzy_pomieszczenia VALUES(2,8);

```

```

INSERT INTO pracownicy_pomieszczenia VALUES(1,10);
INSERT INTO pracownicy_pomieszczenia VALUES(3,2);
INSERT INTO pracownicy_pomieszczenia VALUES(2,3);

```

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

Wyszukanie darmowego zajęcia:

```

SELECT *
FROM Zajecia
WHERE Cena = 0;

```

Wyszukanie numeru sprzętu o kodzie modelu - a48947 :

```

SELECT Nr_sprzetu
FROM sprzety
WHERE Nr_model = (
    SELECT Nr_model
    FROM modele
    WHERE Kod_modelu='a48947'
);

```

Wyszukanie pokoju mieszkalnego o liczbie miejsc większej niż 1:

```

SELECT Nr_pomieszczenia
FROM pokoje_mieszkalne
WHERE Ilosc_miejsc>1;

```

Wyszukanie pomieszczeń znajdujących się na 1 piętrze:

```

SELECT Nr_pomieszczenia
FROM pomieszczenia
WHERE Nr_pietra=1;

```

Zmniejszenie liczby miejsc do dwóch w pokoju mieszkalnym, gdzie liczba miejsc równa 3:

```

UPDATE pokoje_mieszkalne SET ilosc_miejsc=2
WHERE ilosc_miejsc=3;

```