Sprawozdanie z realizacji 1. części projektu z przedmiotu BADA: Bazy Danych i Big Data

Temat: Spółdzielnia mieszkaniowa
Grupa B

Miłosz Kutyła (318427)
Jakub Ossowski (318435)

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska
5 czerwca 2023

Spis treści

Wstęp

Niniejszy dokument to sprawozdanie z realizacji projektu w ramach przedmiotu BADA. Oświadczamy, że ta praca, stanowiąca podstawę do uznania osiągnięcia efektów uczenia się z przedmiotu BADA, została wykonana przez nas samodzielnie.

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

Celem projektu było stworzenie bazy danych dla **spółdzielni mieszkaniowej**. Zadaniem naszej bazy jest gromadzenie danych o **pracownikach**, zarządzanych przez nią **budynkach**, podmiotach powiązanych z **lokalami**. Baza powinna umożliwiać również księgowanie **opłat** czy prowadzenie historii i dokumentacji **zgłoszeń**. Powinna ona posiadać również informacje o **strukturze organizacyjnej** spółdzielni (informacje o aktualnych oraz historycznych członkach zarządu).

Oprogramowanie użyte podczas realizacji projektu: `Toad Data Modeler`, `Toad for Oracle`, `Oracle 21c XE`.

2. Definicja systemu

2.1. Funkcjonalności systemu

Przewidzieliśmy następującą funkcjonalność bazy danych:

1. Dodawanie, modyfikowanie i usuwanie pracowników i ich danych.
2. Podgląd danych pracowników.
3. Dodawanie i modyfikowanie informacji o kadencji członków zarządu.
4. Podgląd informacji o kadencji członków zarządu.
5. Dodawanie, modyfikowanie i usuwanie typów stanowisk oraz ich opisów.
6. Podgląd informacji o stanowisku zajmowanym przez pracownika.
7. Dodawanie, modyfikowanie i usuwanie danych o budynkach, którymi zarządza spółdzielnia.
8. Podgląd danych dotyczących budynków.
9. Dodawanie, modyfikowanie i usuwanie lokali znajdujących się w budynkach.
10. Podgląd danych dotyczących lokali.
11. Dodawanie, modyfikowanie i usuwanie informacji o właścicielach lokali.
12. Podgląd danych dotyczących właścicieli.
13. Dodawanie, modyfikowanie i usuwanie informacji o lokatorach.
14. Podgląd danych dotyczących lokatorów.
15. Dodawanie, modyfikowanie i usuwanie danych o historii i bieżących opłatach nałożonych na lokal.
16. Podgląd historii opłat.
17. Dodawanie, modyfikowanie i usuwanie danych o zgłoszeniach.
18. Podgląd danych dotyczących zgłoszeń.

2.2. Perspektywy użytkowników

Baza spółdzielni mieszkaniowej powinna mieć system kontroli dostępu. W naszym przypadku głównymi użytkownikami bazy danych będą pracownicy spółdzielni. Z tego powodu zdecydowaliśmy się na kontrolę dostępu opartą na rolach, które ściśle odzwierciedlają stanowisko zajmowane przez pracownika. Taki system pozwoli na elastyczne dodawanie i modyfikowanie ról oraz permisji przypisanych do konkretnego stanowiska. Poniżej prezentujemy przykładowe role wraz z permisjami:

- **Administrator**: administruje danymi, ma dostęp do całej bazy i możliwość jej modyfikacji.
- **Prezes/z-ca Prezesa**: ma dostęp do całej bazy danych. Prezes jest w stanie dodawać, modyfikować i usuwać dane dotyczące spółdzielni, pracowników spółdzielni, lokali, pracowników oraz członków zarządu.
- **Członkowie zarządu**: mają wgląd do wszystkich danych. Są w stanie dodawać, modyfikować i usuwać dane dotyczące spółdzielni, budynków, lokali oraz pracowników.
- **Pracownik spółdzielni**: ma dostęp do wszystkich danych z wyłączeniem danych o pracownikach. Jest w stanie dodawać i modyfikować dane dotyczące budynków, lokali, właścicieli, lokatorów, zgłoszeń.
- **Księgowa**: ma dostęp do danych pracowników, lokatorów, właścicieli oraz opłat. Może dodawać, modyfikować i usuwać dane w tabelach do których ma dostęp.
- **Dozorca**: ma wgląd do danych budynków oraz lokali. Jest w stanie dodawać, modyfikować i usuwać dane dotyczące zgłoszeń.

3. Model konceptualny

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

W bazie spółdzielni wyróżniamy następujące encje:

- **Spoldzielnia**: encja główna reprezentująca spółdzielnię.
- **Pracownik**: encja reprezentująca pracownika zatrudnianego przez spółdzielnię.
- **Budynek**: encja reprezentująca budynek zarządzany przez spółdzielnię.
- **Lokal**: encja reprezentująca lokal będący częścią budynku.
- **Lokator**: encja reprezentująca lokatora zamieszkującego lokal.
- **Zgloszenie**: encja reprezentująca zgłoszenie do lokalu, budynku lub spółdzielni.
- **Wlasciciel**: encja reprezentująca właściciela lokalu.
- **Oplata**: encja reprezentująca opłatę za lokal.

Utworzyliśmy również specjalizację **Czlonek zarzadu** encji **Pracownik** żeby wyróżnić pracowników, którzy są członkami zarządu spółdzielni.

3.2. Ustalenie związków między encjami i ich typów

- **Spoldzielnia - Pracownik (1..1 - 0..m)**
  Spółdzielnia zatrudnia pracowników, może być ich wiele, ale istnieje też możliwość, że spółdzielnia nie zatrudnia żadnych pracowników. Każdy pracownik jest zatrudniany tylko przez jedną spółdzielnię.
- **Spoldzielnia - Budynek (1..1 - 0..m)**
  Spółdzielnia zarządza budynkiem, może być ich wiele, ale istnieje też możliwość, że spółdzielnia nie zarządza żadnym budynkiem. Każdy budynek jest zarządzany wyłącznie przez jedną spółdzielnię.
- **Budynek - Lokal (1..1 - 1..m)**
  Budynek posiada lokal, może być ich wiele, ale musi występować minimum jeden. Każdy lokal należy wyłącznie do jednego budynku.
- **Spoldzielnia - Lokal (1..1 - 0..m)**
  Spółdzielnia zarządza lokalem, może być ich wiele, ale istnieje też sytuacja, w której spółdzielnia nie zarządza żadnym lokalem (tożsame z sytuacją, gdy nie zarządza żadnym budynkiem). Lokal może być zarządzany wyłącznie przez jedną spółdzielnię.
- **Budynek - Zgloszenie (0..m - 0..m)**
  Zgłoszenie dotyczy budynku. Jedno zgłoszenie może dotyczyć wielu budynków, ale istnieją też sytuacje, w których zgłoszenie nie dotyczy żadnego budynku. Jeden budynek może posiadać wiele lub 0 zgłoszeń z nim związanych.
- **Spoldzielnia - Zgloszenie (1..1 - 0..m)**
  Spółdzielnia obsługuje zgłoszenie, może być ich wiele, ale istnieje też sytuacja, w której spółdzielnia nie obsługuje żadnych zgłoszeń. Jedno zgłoszenie może być obsługiwane wyłącznie przez jedną spółdzielnie.
- **Zgloszenie - Lokal (0..m - 0..m)**
  Zgłoszenie dotyczy lokalu. Jedno zgłoszenie może dotyczyć wielu lokali, ale istnieją też sytuacje, w których zgłoszenie nie dotyczy żadnego lokalu. Jeden lokal może posiadać wiele lub 0 zgłoszeń z nim związanych.
- **Lokal - Lokator (1..m - 0..m)**
  Lokal posiada lokatora. Lokal może posiadać wielu lokatorów, ale istnieje też sytuacja, w której lokal nie posiada żadnego lokatora. Lokator może zamieszkiwać wiele lokali, ale jeśli istnieje to musi zamieszkiwać minimum 1 lokal.
- **Lokal - Wlasciciel (1..m - 0..m)**
  Lokal posiada właściciela. Lokal może posiadać wielu właścicieli, ale istnieje też sytuacja, w której lokal nie posiada żadnego właściciela (np. nie został jeszcze sprzedany, właścicielem jest spółdzielnia). Właściciel może posiadać wiele lokali, ale jeśli istnieje to musi posiadać minimum 1 lokal.
- **Lokal - Oplata (1..1 - 0..m)**
  Opłaty nałożone na lokal. Na lokal może być nałożone wiele opłat, ale istnieje też sytuacja, w której na lokal nie są nałożone żadne opłaty (np. lokal nie jest jeszcze ukończony, nie posiada właściciela lub najemcy). Pojedyncza opłata może być nałożona jedynie na jeden lokal.

3.3. Określenie atrybutów i ich dziedzin

- Spoldzielnia:

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_spoldzielni	SmallInt	Obowiązkowy, klucz główny pozwalający na identyfikację spółdzielni	Unikatowy numer spółdzielni
Nazwa	VarChar(100)	Obowiązkowy	Nazwa spółdzielni
Adres	VarChar(400)	Obowiązkowy	Adres spółdzielni, pole segmentowe zawiera informacje o mieście, ulicy i numerze lokalu
Prezes	VarChar(50)	Obowiązkowy	Prezes spółdzielni
Data_zalozenia	Date	Obowiązkowy	Data założenia spółdzielni

- Pracownik:

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_pracownika	Integer	Obowiązkowy, klucz główny pozwalający na identyfikację pracownika	Unikatowy identyfikator pracownika
Imie	VarChar(20)	Obowiązkowy	Imię pracownika
Nazwisko	VarChar(30)	Obowiązkowy	Nazwisko pracownika
Data_urodzenia	Date	Obowiązkowy	Data urodzenia
PESEL	Character(11)	Nieobowiązkowy	PESEL
Plec	PlecD	Obowiązkowy	Płeć pracownika
Adres	VarChar(400)	Obowiązkowy	Adres pracownika, pole segmentowe zawiera informacje o mieście, ulicy i numerze lokalu
Stanowisko	VarChar(20)	Obowiązkowy	Stanowisko
Data_zatrudnienia	Date	Obowiązkowy	Data zatrudnienia
Wynagrodzenie	Money	Obowiązkowy	Wynagrodzenie pracownika
Nr_konta	Character(26)	Nieobowiązkowy	Numer konta
Email	VarChar(30)	Nieobowiązkowy	E-mail pracownika
Nr_telefonu	VarChar(15)	Nieobowiązkowy	Numer telefonu pracownika

- Budynek:

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_budynku	Integer	Obowiązkowy, klucz główny pozwalający na identyfikację budynku	Numer identyfikujący budynek
Adres	VarChar(400)	Obowiązkowy	Adres budynku, pole segmentowe zawiera informacje o mieście, ulicy, numerze lokalu, kodzie pocztowym oraz poczcie
Powierzchnia	Float	Obowiązkowy	Powierzchnia budynku określona w metrach kwadratowych
Rodzaj	VarChar(50)	Obowiązkowy	Rodzaj budynku
Wykonawca	VarChar(300)	Obowiązkowy	Generalny wykonawca budynku
Liczba_kondygnacji	SmallInt	Nieobowiązkowy	Liczba kondygnacji budynku
Liczba_lokali	SmallInt	Obowiązkowy	Liczba lokali w budynku
Liczba_skladzikow	SmallInt	Nieobowiązkowy	Liczba składzików w budynku

- Lokal:

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_lokalu	Integer	Obowiązkowy, klucz główny pozwalający na identyfikację lokalu	Unikatowy numer identyfikujący lokal
Numer	SmallInt	Nieobowiązkowy	Numer lokalu w danym budynku mieszkalnym
Cena_najmu	Money	Nieobowiązkowy	Cena najmu mieszkania
Powierzchnia	Float	Obowiązkowy	Powierzchnia lokalu wyrażona w metrach kwadratowych
Liczba_pokojow	SmallInt	Obowiązkowy	Liczba pokojów
Pietro	SmallInt	Obowiązkowy	Numer piętra, na którym znajduje się lokal

- Zgloszenie:

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_zgloszenia	Integer	Obowiązkowy, klucz główny pozwalający na identyfikację zgłoszenia	Numer identyfikujący zgłoszenie
Data_zgloszenia	Date	Obowiązkowy	Data zgłoszenia
Data_zamkniecia	Date	Nieobowiązkowy	Data zamknięcia zgłoszenia
Typ	VarChar(300)	Obowiązkowy	Typ zgłoszenia
Opis	VarChar(500)	Nieobowiązkowy	Krótki opis zgłoszenia
Koszt	Money	Nieobowiązkowy	Sumaryczny koszt rozpatrzenia i rozwiązania zgłoszenia

- Lokator:

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_lokatora	Integer	Obowiązkowy, klucz główny pozwalający na identyfikację lokatora	Unikalny numer identyfikujący lokatora
Imie	VarChar(20)	Obowiązkowy	Imię lokatora
Nazwisko	VarChar(30)	Obowiązkowy	Nazwisko lokatora
Data_urodzenia	Date	Obowiązkowy	Data urodzenia lokatora
PESEL	Character(11)	Nieobowiązkowy	Numer PESEL lokatora
Plec	PlecD	Obowiązkowy	Płeć lokatora
Email	VarChar(30)	Nieobowiązkowy	E-mail lokatora
Nr_telefonu	VarChar(15)	Nieobowiązkowy	Numer telefonu lokatora

- Wlasciciel:

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_wlasciciela	Integer	Obowiązkowy, klucz główny pozwalający na identyfikację właściciela	Unikalny numer identyfikujący właściciela
Imie	VarChar(20)	Obowiązkowy	Imię właściciela
Nazwisko	VarChar(30)	Obowiązkowy	Nazwisko właściciela
Data_urodzenia	Date	Obowiązkowy	Data urodzenia
PESEL	Character(11)	Nieobowiązkowy	Numer PESEL właściciela
Plec	PlecD	Obowiązkowy	Plec właściciela
Adres	VarChar(400)	Obowiązkowy	Adres budynku, pole segmentowe zawiera informacje o mieście, ulicy, numerze lokalu, kodzie pocztowym oraz poczcie
Email	VarChar(30)	Nieobowiązkowy	E-mail właściciela
Nr_telefonu	VarChar(15)	Nieobowiązkowy	Numer telefonu właściciela

- Oplata:

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Nr_oplaty	Integer	Obowiązkowy, klucz główny pozwalający na identyfikację opłaty	Numer identyfikujący opłatę
Data	Date	Obowiązkowy	Data wystawienia zadania
Wysokosc	Money	Obowiązkowy	Wysokość opłaty
Nr_konta	Character(26)	Nieobowiązkowy	Numer konta bankowego, na które należy uiścić opłatę
Status	StatusD	Obowiązkowy	Aktualny status opłaty

- Czlonek zarzadu:

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Od_kiedy	Date	Obowiązkowy	Data od kiedy pracownik jest członkiem zarządu
Do_kiedy	Date	Obowiązkowy	Data do kiedy pracownik będzie członkiem zarządu
Rola	ZarzadD	Obowiązkowy	Rola jaką członek pełni w zarządzie

Dziedziny

Na poziomie modelu konceptualnego stworzyliśmy trzy dziedziny dostosowane do naszych potrzeb:

- **PlecD**: dziedzina utworzona na podstawie typu Character(1) z dopuszczalnymi wartościami "K" (kobieta) oraz "M" (mężczyzna). Jest wykorzystana w polach "Plec" w encjach reprezentujących osoby: "Pracownik", "Lokator", "Wlasciciel".
- **StatusD**: dziedzina utworzona na podstawie typu VarChar(11) z dopuszczalnymi wartościami "Oplacono" oraz "Nieoplacono". Jest wykorzystana w polu "Status" encji "Oplata".
- **ZarzadD**: dziedzina utworzona na podstawie typu VarChar(15) z dopuszczalnymi wartościami "Prezes", "Z-ca Prezesa" oraz "Członek zarządu". Jest wykorzystana w polu "Rola" w specjalizacji "Czlonek_zarzadu".

3.4. Dodatkowe reguły integralnościowe (reguły biznesowe)

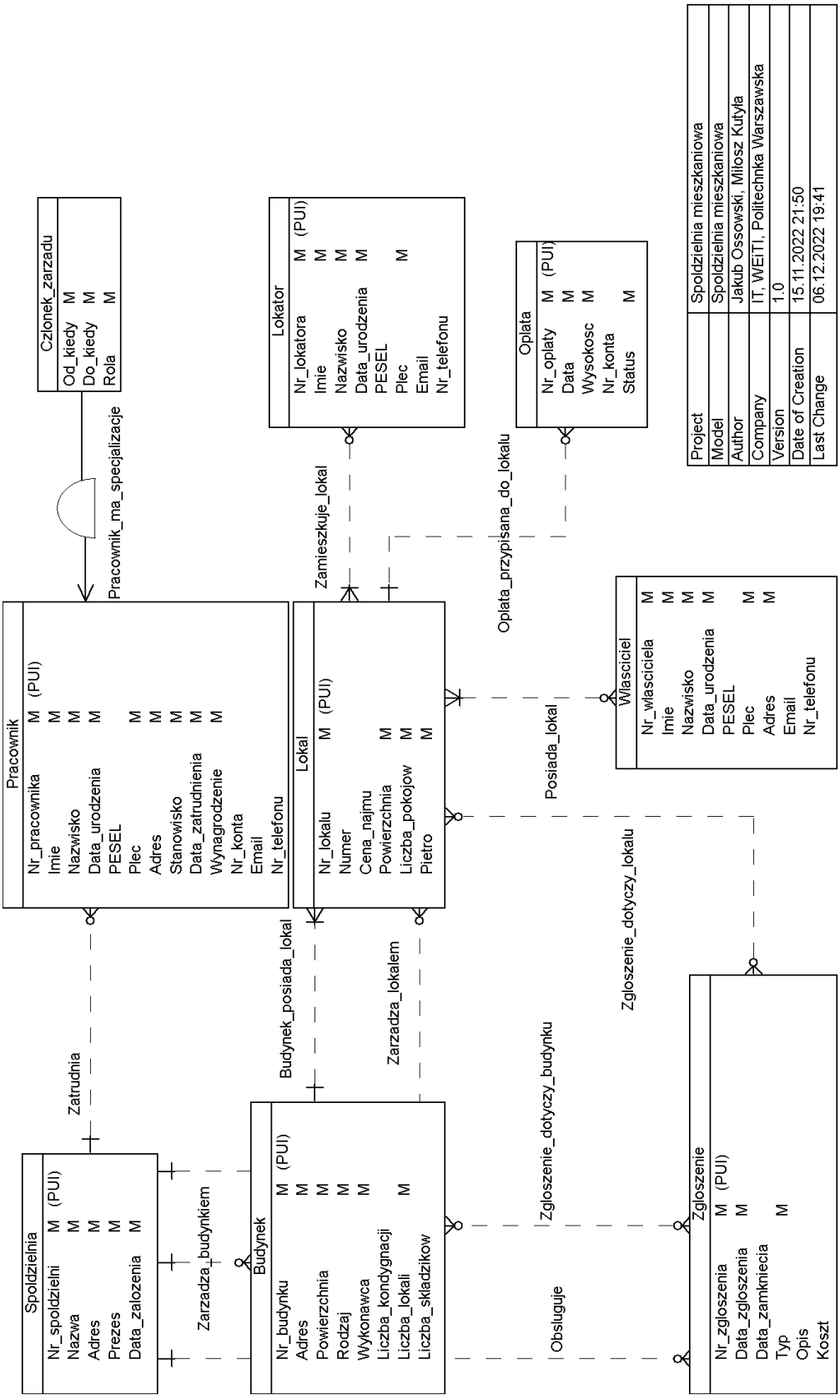
Reguły integralnościowe to zbiór reguł określających, które stany bazy danych są poprawne, czyli jakie operacje prowadzące do modyfikacji danych są dozwolone. Podczas tworzenia modelu konceptualnego zadbaliśmy o to, żeby zawartości pól rekordów były zgodne z określonym typem (lub dziedziną). Wymagania obecności danych została przedstawiona w poprzedniej sekcji. Dodatkowo zakładamy, że Pracownik może pracować tylko na jednym stanowisku (ewentualnie mieć dodatkową rolę w zarządzie).

3.5. Klucze kandydujące i główne

W naszym modelu zastosowaliśmy klucze sztuczne, czyli sztucznie wygenerowane numery identyfikujące. Takie rozwiązanie poprawia czytelność bazy oraz zwiększa jej szybkość działania. W naszym przypadku, w niektórych encjach nie występowały inne klucze kandydujące lub klucze kandydujące miały większy zakres wartości niż klucze sztuczne.

Encja	**Klucz Główny**	**Klucze kandydujące**
Spoldzielnia	Nr_spoldzielni	
Pracownik	Nr_pracownika	PESEL
Budynek	Nr_budynku	Adres
Lokal	Nr_lokalu	
Lokator	Nr_lokatora	PESEL
Oplata	Nr_oplaty	
Wlasciciel	Nr_wlasciciela	PESEL
Zgloszenie	Nr_zgloszenia	

3.6. Schemat ER na poziomie konceptualnym

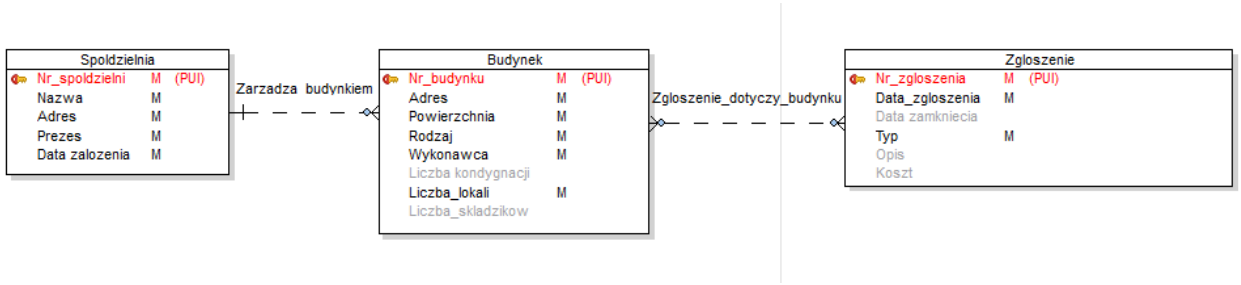


3.7. Problem pułapek szczelinowych i wachlarzowych - analiza i przykłady

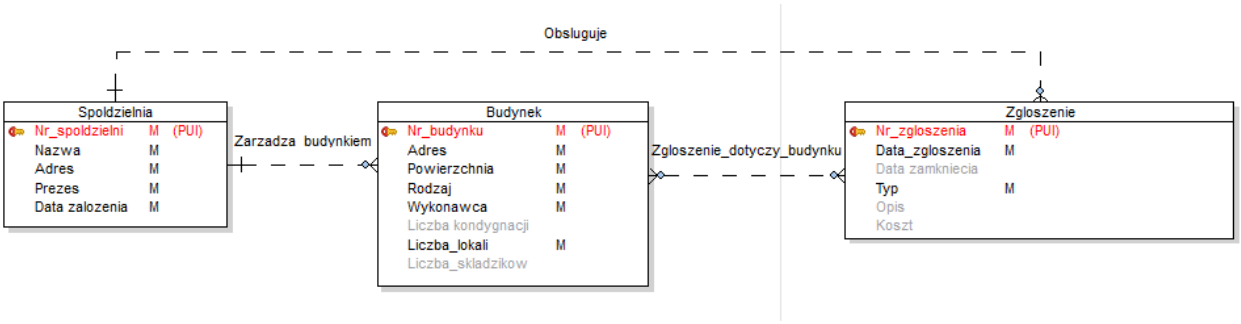
W procesie projektowania nie stworzyliśmy żadnej pułapki, poniżej przedstawiamy przykładowe pułapki, którym udało nam się zapobiec.

Szczelinowa

Każde zgłoszenie jest powiązane z budynkiem, po usunięciu budynku tracimy informację o historycznym zgłoszeniu.

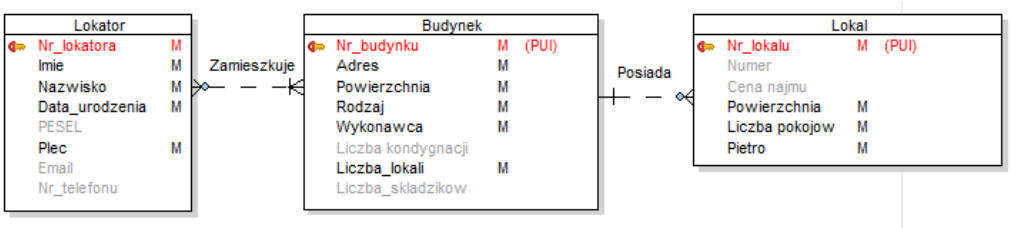


Rozwiązanie: dodanie relacji pomiędzy zgłoszeniem a spółdzielnią



Wachlarzowa

Lokator jest powiązany z budynkiem, a lokal z budynkiem. Sytuacja np. domku jednorodzinnego. Tracimy wówczas informację jaki lokal jest zamieszkiwany przez lokatora w budynkach, które posiadają wiele lokali.



Rozwiązanie: powiązanie lokatora z lokalem i lokalu z budynkiem oraz przyjęcie założenia, że budynek musi posiadać minimum jeden lokal (dom jednorodzinny to budynek z jednym lokalem).

4. Model logiczny

4.1. Charakterystyka modelu relacyjnego

W celu przekształcenia modelu konceptualnego w model relacyjny, na początku skorzystaliśmy z automatycznej konwersji programu `Toad Data Modeler`. Dla każdej encji modelu konceptualnego tworzy odpowiadającą jej tabele. Program pozbywa się wszystkich związków wielu do wielu. Aby to zrealizować zastępuje je tabelami skrzyżowań, w których klucz podstawowy składa się z kluczy obcych z tabel, które były powiązane związkiem wiele do wielu. Przekształca niektóre typy danych, na takie występujące w wybranej bazie danych np. typ `Money` na `Number` lub `Mandatory` na `NOTNULL`. Dodaje on również klucze obce do odpowiednich związków. Następnie przeszliśmy do procesów usunięcia niekompatybilności oraz normalizacji, które dokładniej opisaliśmy w dalszych punktach.

4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

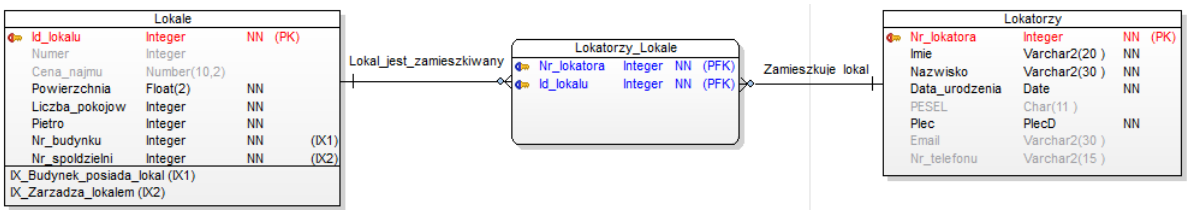
4.2.1. Nazewnictwo

Encje: W modelu logicznym poszczególne encje zamieniają się w tabele, które reprezentują całe zbiory obiektów w przeciwieństwie do modelu konceptualnego, gdzie encje reprezentują pojedynczy obiekt. Wobec czego dostosowaliśmy nazwy tabel zmieniając je na liczbę mnogą.

Związki: Niektóre nazwy związków zaczęły się powtarzać, a niektóre stały się nieintuicyjne lub nie pasowały do przyjętej konwencji np. związki tabel skrzyżowań (nazwę *Zgloszenie_dotyczy_budynku_Budynek* i *Zgloszenie_dotyczy_budynku_Zgloszenie*, zamieniliśmy na *Zglosznie_dotyczy_budynku* i *Budynek_ma_zgloszenie*). W podobny sposób zmieniliśmy nazwy innych związków, tak aby nasz model był spójny i czytelny.

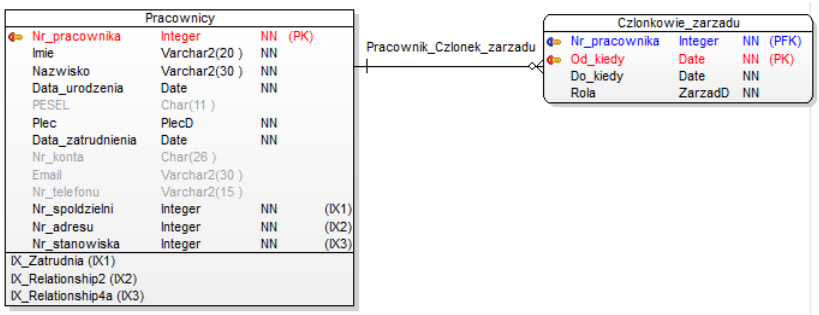
4.2.2. Związki wiele do wielu

W naszym modelu konceptualnym występowało wiele związków wiele do wielu. Takie związki są niekompatybilne z modelem logicznym. W celu zastąpienia tych związków zastosowaliśmy tablice skrzyżowań (ang. *bridge tables*). W naszym przypadku program `Toad Data Modeler` skutecznie zastąpił takie związki automatycznie i z naszej strony wiązało się to jedynie z weryfikacją i dostosowaniem nazewnictwa. Poniżej przykład takiego rozwiązania:



4.2.3. Specjalizacje

Usunęliśmy wszystkie specjalizacje (w naszym przypadku była to jedna specjalizacja *Pracownika*) i zastąpiliśmy je odpowiednimi relacjami, ponieważ specjalizacje nie występują w modelu relacyjnym. Poniżej dostosowanie naszej relacji:



4.3. Proces normalizacji – analiza i przykłady

4.3.1. Przejście do pierwszej postaci normalnej

Warunki osiągnięcia 1PN przez rozpatrywaną relację:

1. każda wartość atrybutu w każdej krotce relacji jest wartością elementarną,
2. relacja nie ma powtarzających się grup.

Pierwszy warunek nie był spełniony przez atrybut Adres występujący w kilku relacjach naszego modelu. W celu rozwiązania tego problemu utworzyliśmy relację Adresy z atrybutami: Nr_adresu, Ulica, Nr_budynku, Nr_lokalu oraz Miasto. Aby zlikwidować powtarzające się w bazie atrybuty usunęliśmy atrybut Adres z relacji: Spoldzielnie, Pracownicy, Budynki, Lokale, Wlasciciele. Każdą z tych relacji następnie połączyliśmy związkiem 0..1-1..1 z relacją Adresy.
Problemem okazał się również atrybut Prezes w relacji Spoldzielnie niebędący polem elementarnym, ponieważ składał się z imienia i nazwiska. Wprowadzał on również redundancję, ponieważ tylko jeden Pracownik (w danej spółdzielni) ma rolę Prezes (specjalizacja). W związku z tym zdecydowaliśmy się usunąć ten atrybut.
Następne poprawki były związane z 2. warunkiem osiągnięcia 1PN przez relację. Wprowadziliśmy relację Wykonawcy zastępującą atrybut Wykonawca w relacji Budynki, aby uniknąć sytuacji, w której dany wykonawca występuje w wielu budynkach. Jej atrybuty to Nr_wykonawcy, Nazwa, Opis oraz Nr_adresu (klucz obcy z relacji Adresy). Relację Wykonawcy połączyliśmy związkiem 0..1-0..n z relacją Budynki oraz związkiem 0..1-1..1 z relacją Adresy.
Wprowadziliśmy relację Rodzaje_budynkow zastępującą atrybut Rodzaj w relacji Budynki, aby uniknąć sytuacji, w której dany rodzaj występuje w wielu budynkach. Jej atrybuty to Nr_rodzaju, Rodzaj oraz Opis. Połączyliśmy ją związkiem 1..1-0..n z relacją Budynki.
Wprowadziliśmy relację Typ_zgloszenia zastępującą atrybut Typ w relacji Zgloszenia, aby uniknąć sytuacji, w której dany typ występuje w wielu zgłoszeniach. Jego atrybuty to Nr_typu, Typ oraz Opis. Połączyliśmy ją związkiem 1..1-0..n z relacją Zgloszenia.
Wprowadziliśmy relację Stanowiska zastępujących atrybut Stanowisko w relacji Pracownicy, aby uniknąć sytuacji, w której dane stanowisko występuje dla wielu pracowników. Jej atrybuty to Nr_stanowiska, Nazwa_stanowiska oraz Opis. Połączyliśmy ją związkiem 1..1-0..n z relacją Pracownicy.
Po zmianach utworzony model był w pierwszej postaci normalnej.

4.3.2. Przejście do drugiej postaci normalnej

Warunki osiągnięcia 2PN przez rozpatrywaną relację:

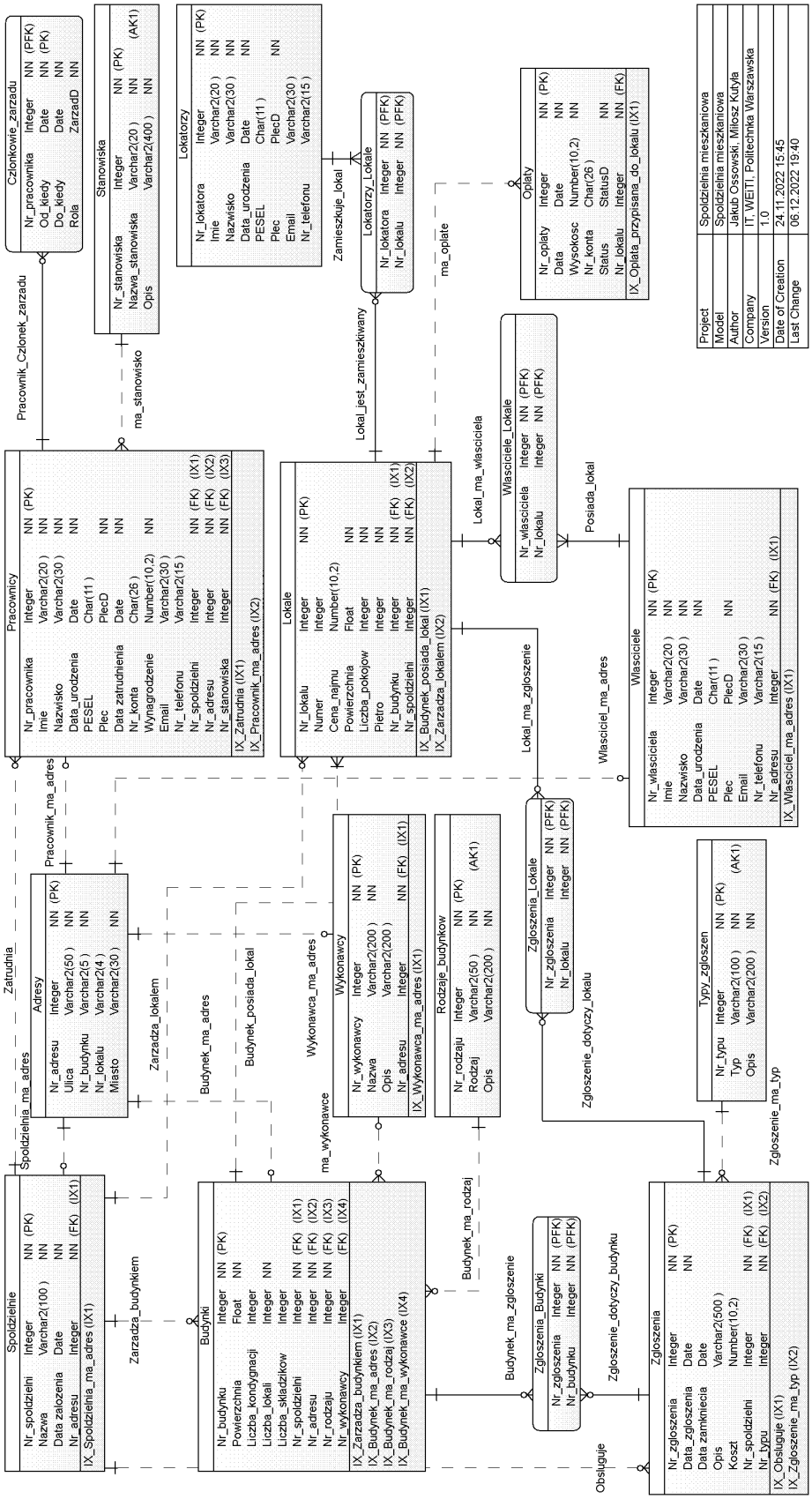
1. relacja jest w 1PN,
2. każdy atrybut relacji niewchodzący w skład żadnego klucza potencjalnego jest w pełni funkcyjnie zależny od wszystkich kluczy potencjalnych tej relacji,
3. każdy atrybut relacji niewchodzący w skład klucza zależy od klucza a nie od jego części.

Równoważnie relacja będąca w 1PN jest w 2PN, jeśli wszystkie klucze potencjalne relacji są kluczami prostymi. Ze względu na rozbicie klucza kandydującego Adres w relacji Budynki (patrz 3.5) przy osiąganiu 1PN, wszystkie klucze kandydujące są kluczami prostymi. Z tego powodu nasz model miał już 2PN.

4.3.3. Przejście do trzeciej postaci normalnej

Trzecia postać normalna zakłada, że relacja jest w 2PN, oraz że wszystkie niekluczowe kolumny są określone kluczem, całym kluczem i tylko kluczem. Problemem była relacja Czlonkowie_zarzadu - konieczne było skonstruowanie klucza głównego z atrybuty Od_kiedy oraz klucza obcego Nr_pracownika, aby zapewnić unikalność każdej krotki (ten sam pracownik może być np. prezesem przez dwie kadencje).
Ostatnią relacją, w której atrybut mógłby powodować, że relacja nie jest w 3PN, było Wynagrodzenie w relacji Pracownicy i jego teoretyczna zależność od atrybutu Nr_stanowiska. Po analizie problemu zdecydowaliśmy się na to, żeby nie wprowadzać nowej tabeli Wynagrodzenia dla poszczególnych stanowisk. Doszliśmy do wniosku, że osoby na jednakowych stanowiskach *mogą* mieć różne wynagrodzenia ze względu na staż w spółdzielni lub ze względu na pełnienie dodatkowej funkcji jako członek zarządu.
Po zmianach utworzony model był w trzeciej postaci normalnej.

4.4. Schemat ER na poziomie logicznym



4.5. Więzy integralności

Do zapewnienia integralności bazy upewniliśmy się, że nie występują żadne pola segmentowe - wszystkie wyeliminowaliśmy przy normalizacji (również zajmowanie wielu stanowisk przez jednego Pracownika). W atrybutach, które są niezbędne do działania bazy, zaznaczyliśmy opcję `NOT NULL`, a w tabelach słownikowych wartości określające unikalny typ (rodzaj, nazwę) oznaczyliśmy jako `UNIQUE`. Wszystkie klucze główne i pochodne mają również zapewnioną unikalność.

4.6. Proces denormalizacji – analiza i przykłady

W ramach denormalizacji, czyli procesu zmiany schematu relacji, w efekcie której stopień normalizacji relacji jest mniejszy od stopnia normalizacji co najmniej jednej z relacji oryginalnych w celu poprawy wydajności systemu, rozważaliśmy odrzucenie tabel słownikowych (typy, rodzaje). Ostatecznie nie zdecydowaliśmy się na to, ponieważ uważamy, że każdy typ wymagał dodatkowego opisu. W związku z tym w naszym modelu nie zdecydowaliśmy się na denormalizację.

5. Faza fizyczna

5.1. Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Wymagane zasoby	Czy wykonalne?	Uwagi
Dodawanie, modyfikowanie i usuwanie pracowników i ich danych	Spoldzienie, Pracownicy, Adresy, Stanowiska	tak	brak
Podgląd danych (personalnych) pracowników	Pracownicy, Adresy	tak	brak
Dodawanie i modyfikowanie informacji o kadencji członków zarządu	Pracownicy, Czlonkowie_zarzadu	tak	brak
Podgląd informacji o kadencji członków zarządu	Pracownicy, Czlonkowie_zarzadu	tak	brak
Dodawanie, modyfikowanie i usuwanie typów stanowisk oraz ich opisów	Stanowiska	tak	brak
Podgląd informacji o stanowisku zajmowanym przez pracownika	Pracownicy, Stanowiska	tak	brak
Dodawanie, modyfikowanie i usuwanie danych o budynkach, którymi zarządza spółdzielnia	Spoldzielnie, Budynki, Rodzaje_budynkow, Adresy	tak	brak
Podgląd danych dotyczących budynków	Budynki, Wykonawcy, Rodzaje_budynkow, Adresy	tak	brak
Dodawanie, modyfikowanie i usuwanie lokali znajdujących się w budynkach	Budynki, Lokale	tak	brak
Podgląd danych dotyczących lokali	Lokale, Budynki, Adresy	tak	brak
Dodawanie, modyfikowanie i usuwanie informacji o właścicielach lokali	Wlasciciele, Wlasciciele_Lokale, Adresy	tak	brak
Podgląd danych dotyczących właścicieli	Wlasciciele, Wlasciciele_Lokale, Adresy	tak	brak
Dodawanie, modyfikowanie i usuwanie informacji o lokatorach	Lokatorzy, Lokatorzy_Lokale	tak	brak
Dodawanie, modyfikowanie i usuwanie danych o historii i bieżących opłatach nałożonych na lokal	Oplaty, Lokale	tak	brak
Podgląd historii opłat	Oplaty	tak	brak
Dodawanie, modyfikowanie i usuwanie danych o zgłoszeniach	Spoldzielnie, Zgloszenia, Typ_zgloszen	tak	brak

Podgląd danych dotyczących zgłoszeń	Zgloszenia, Typ_zgloszen	tak	brak
Podgląd zgłoszeń dot. lokali	Zgloszenia, Typ_zgloszen, Zgloszenia_Lokale, Lokale	tak	brak
Podgląd zgłoszeń dot. budynków	Zgloszenia, Typ_zgloszen, Zgloszenia_Budynki, Budynki	tak	brak

5.2. Strojenie bazy danych – dobór indeksów

Spoldzielnie - adres
```
CREATE INDEX IX_Spoldzielnia_ma_adres ON Spoldzielnie (Nr_adresu)
```

Pracownicy - spółdzielnia, adres, stanowisko
```
CREATE INDEX IX_Zatrudnia ON Pracownicy (Nr_spoldzielni)
CREATE INDEX IX_Pracownik_ma_adres ON Pracownicy (Nr_adresu)
CREATE INDEX IX_Pracownik_ma_stanowisko ON Pracownicy (Nr_stanowiska)
```

Budynki - spółdzielnia, adres, rodzaj, wykonawcy
```
CREATE INDEX IX_Zarzadza_budynkiem ON Budynki (Nr_spoldzielni)
CREATE INDEX IX_Budynek_ma_adres ON Budynki (Nr_adresu)
CREATE INDEX IX_Budynek_ma_rodzaj ON Budynki (Nr_rodzaju)
CREATE INDEX IX_Budynek_ma_wykonawce ON Budynki (Nr_wykonawcy)
```

Lokale - budynek, spółdzielnia
```
CREATE INDEX IX_Budynek_posiada_lokal ON Lokale (Nr_budynku)
CREATE INDEX IX_Zarzadza_lokalem ON Lokale (Nr_spoldzielni)
```

Zgloszenia - spółdzielnia, typ
```
CREATE INDEX IX_Obsluguje ON Zgloszenia (Nr_spoldzielni)
CREATE INDEX IX_Zgloszenie_ma_typ ON Zgloszenia (Nr_typu)
```

Wlasciciele - adres
```
CREATE INDEX IX_Wlasciciel_ma_adres ON Wlasciciele (Nr_adresu)
```

Oplaty - lokal
```
ALTER TABLE Oplaty ADD CONSTRAINT PK_Oplaty PRIMARY KEY (Nr_oplaty)
```

5.3. Skrypt SQL zakładający bazę danych

```
/*
Created: 24.11.2022
Modified: 06.12.2022
Project: Spoldzielnia mieszkaniowa
Model: Spoldzielnia mieszkaniowa
Company: IT, WEiTI, Politechnka Warszawska
Author: Jakub Ossowski, Miłosz Kutyła
Version: 1.0
Database: Oracle 19c
*/


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

-- Table Spoldzielnie
```

```
CREATE TABLE Spoldzielnie(
  Nr_spoldzielni Integer NOT NULL,
  Nazwa Varchar2(100 ) NOT NULL,
  Data_zalozenia Date NOT NULL,
  Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Spoldzielnie

CREATE INDEX IX_Spoldzielnia_ma_adres ON Spoldzielnie (Nr_adresu)
/

-- Add keys for table Spoldzielnie

ALTER TABLE Spoldzielnie ADD CONSTRAINT PK_Spoldzielnia PRIMARY KEY (Nr_spoldzielni)
/

-- 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 ) NOT NULL
        CHECK (Plec IN ('K','M')),
  Data_zatrudnienia Date NOT NULL,
  Nr_konta Char(26 ),
  Email Varchar2(30 ),
  Nr_telefonu Varchar2(15 ),
  Nr_spoldzielni Integer NOT NULL,
  Nr_adresu Integer NOT NULL,
  Nr_stanowiska Integer NOT NULL
)
/

-- Create indexes for table Pracownicy

CREATE INDEX IX_Zatrudnia ON Pracownicy (Nr_spoldzielni)
/

CREATE INDEX IX_Pracownik_ma_adres ON Pracownicy (Nr_adresu)
/

CREATE INDEX IX_Pracownik_ma_stanowisko ON Pracownicy (Nr_stanowiska)
/

-- Add keys for table Pracownicy

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

-- Table Budynki
```

```
CREATE TABLE Budynki(
  Nr_budynku Integer NOT NULL,
  Powierzchnia Float NOT NULL,
  Liczba_kondygnacji Integer,
  Liczba_lokali Integer NOT NULL,
  Liczba_skladzikow Integer,
  Nr_spoldzielni Integer NOT NULL,
  Nr_adresu Integer NOT NULL,
  Nr_rodzaju Integer NOT NULL,
  Nr_wykonawcy Integer
)
/

-- Create indexes for table Budynki

CREATE INDEX IX_Zarzadza_budynkiem ON Budynki (Nr_spoldzielni)
/

CREATE INDEX IX_Budynek_ma_adres ON Budynki (Nr_adresu)
/

CREATE INDEX IX_Budynek_ma_rodzaj ON Budynki (Nr_rodzaju)
/

CREATE INDEX IX_Budynek_ma_wykonawce ON Budynki (Nr_wykonawcy)
/

-- Add keys for table Budynki

ALTER TABLE Budynki ADD CONSTRAINT PK_Budynku PRIMARY KEY (Nr_budynku)
/

-- Table Lokale

CREATE TABLE Lokale(
  Nr_lokalu Integer NOT NULL,
  Numer Integer,
  Cena_najmu Number(10,2),
  Powierzchnia Float NOT NULL,
  Liczba_pokojow Integer NOT NULL,
  Pietro Integer NOT NULL,
  Nr_budynku Integer NOT NULL,
  Nr_spoldzielni Integer NOT NULL
)
/

-- Create indexes for table Lokale

CREATE INDEX IX_Budynek_posiada_lokal ON Lokale (Nr_budynku)
/

CREATE INDEX IX_Zarzadza_lokalem ON Lokale (Nr_spoldzielni)
/

-- Add keys for table Lokale
```

```
ALTER TABLE Lokale ADD CONSTRAINT PK_Lokalu PRIMARY KEY (Nr_lokalu)
/

-- Table Zgloszenia

CREATE TABLE Zgloszenia(
  Nr_zgloszenia Integer NOT NULL,
  Data_zgloszenia Date NOT NULL,
  Data_zamkniecia Date,
  Opis Varchar2(500 ),
  Koszt Number(10,2),
  Nr_spoldzielni Integer NOT NULL,
  Nr_typu Integer NOT NULL
)
/

-- Create indexes for table Zgloszenia

CREATE INDEX IX_Obsluguje ON Zgloszenia (Nr_spoldzielni)
/

CREATE INDEX IX_Zgloszenie_ma_typ ON Zgloszenia (Nr_typu)
/

-- Add keys for table Zgloszenia

ALTER TABLE Zgloszenia ADD CONSTRAINT PK_Zgloszenia PRIMARY KEY (Nr_zgloszenia)
/

-- Table Lokatorzy

CREATE TABLE Lokatorzy(
  Nr_lokatora Integer NOT NULL,
  Imie Varchar2(20 ) NOT NULL,
  Nazwisko Varchar2(30 ) NOT NULL,
  Data_urodzenia Date NOT NULL,
  PESEL Char(11 ),
  Plec Char(1 ) NOT NULL
        CHECK (Plec IN ('K','M')),
  Email Varchar2(30 ),
  Nr_telefonu Varchar2(15 )
)
/

-- Add keys for table Lokatorzy

ALTER TABLE Lokatorzy ADD CONSTRAINT PK_Lokatora PRIMARY KEY (Nr_lokatora)
/

-- Table Wlasciciele

CREATE TABLE Wlasciciele(
  Nr_wlasciciela Integer NOT NULL,
  Imie Varchar2(20 ) NOT NULL,
  Nazwisko Varchar2(30 ) NOT NULL,
  Data_urodzenia Date NOT NULL,
```

```
  PESEL Char(11 ),
  Plec Char(1 ) NOT NULL
        CHECK (Plec IN ('K','M')),
  Email Varchar2(30 ),
  Nr_telefonu Varchar2(15 ),
  Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Wlasciciele

CREATE INDEX IX_Wlasciciel_ma_adres ON Wlasciciele (Nr_adresu)
/

-- Add keys for table Wlasciciele

ALTER TABLE Wlasciciele ADD CONSTRAINT PK_Wlasciciel PRIMARY KEY (Nr_wlasciciela)
/

-- Table Oplaty

CREATE TABLE Oplaty(
  Nr_oplaty Integer NOT NULL,
  Data Date NOT NULL,
  Wysokosc Number(10,2) NOT NULL,
  Nr_konta Char(26 ),
  Status Char(11 ) NOT NULL
        CHECK (Status IN ('Oplacono','Nieoplacono')),
  Nr_lokalu Integer NOT NULL
)
/

-- Create indexes for table Oplaty

CREATE INDEX IX_Oplata_przypisana_do_lokalu ON Oplaty (Nr_lokalu)
/

-- Add keys for table Oplaty

ALTER TABLE Oplaty ADD CONSTRAINT PK_Oplaty PRIMARY KEY (Nr_oplaty)
/

-- Table Czlonkowie_zarzadu

CREATE TABLE Czlonkowie_zarzadu(
  Nr_pracownika Integer NOT NULL,
  Od_kiedy Date NOT NULL,
  Do_kiedy Date NOT NULL,
  Rola Char(256 ) NOT NULL
        CHECK (Rola IN ('Prezes', 'Z-ca Prezesa', 'Członek Zarządu'))
)
/

-- Add keys for table Czlonkowie_zarzadu

ALTER TABLE Czlonkowie_zarzadu ADD CONSTRAINT Unique_Identifier11 PRIMARY KEY (Nr_pracownika,Od_kiedy)
```

```
/

-- Table Lokatorzy_Lokale

CREATE TABLE Lokatorzy_Lokale(
  Nr_lokatora Integer NOT NULL,
  Nr_lokalu Integer NOT NULL
)
/

-- Table Wlasciciele_Lokale

CREATE TABLE Wlasciciele_Lokale(
  Nr_wlasciciela Integer NOT NULL,
  Nr_lokalu Integer NOT NULL
)
/

-- Table Zgloszenia_Budynki

CREATE TABLE Zgloszenia_Budynki(
  Nr_zgloszenia Integer NOT NULL,
  Nr_budynku Integer NOT NULL
)
/

-- Table Zgloszenia_Lokale

CREATE TABLE Zgloszenia_Lokale(
  Nr_zgloszenia Integer NOT NULL,
  Nr_lokalu Integer NOT NULL
)
/

-- Table Adresy

CREATE TABLE Adresy(
  Nr_adresu Integer NOT NULL,
  Ulica Varchar2(50 ) NOT NULL,
  Nr_budynku Varchar2(5 ) NOT NULL,
  Nr_lokalu Varchar2(4 ),
  Miasto Varchar2(30 ) NOT NULL
)
/

-- 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.Ulica IS 'Ulica'
/
```

```
COMMENT ON COLUMN Adresy.Nr_budynku IS 'Numer budynku
'
/
COMMENT ON COLUMN Adresy.Nr_lokalu IS 'Numer lokalu
'
/
COMMENT ON COLUMN Adresy.Miasto IS 'Miasto'
/

-- 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_stanowiska UNIQUE (Nazwa_stanowiska)
/

-- Table Rodzaje_budynkow

CREATE TABLE Rodzaje_budynkow(
  Nr_rodzaju Integer NOT NULL,
  Rodzaj Varchar2(50 ) NOT NULL,
  Opis Varchar2(200 ) NOT NULL
)
/

-- Add keys for table Rodzaje_budynkow

ALTER TABLE Rodzaje_budynkow ADD CONSTRAINT PK_Rodzaje_budynkow PRIMARY KEY (Nr_rodzaju)
/

ALTER TABLE Rodzaje_budynkow ADD CONSTRAINT Rodzaj UNIQUE (Rodzaj)
/

-- Table and Columns comments section

COMMENT ON COLUMN Rodzaje_budynkow.Rodzaj IS 'Rodzaj budynku'
/
COMMENT ON COLUMN Rodzaje_budynkow.Opis IS 'Opis rodzaju budynku'
/

-- Table Typy_zgloszen

CREATE TABLE Typy_zgloszen(
  Nr_typu Integer NOT NULL,
  Typ Varchar2(100 ) NOT NULL,
  Opis Varchar2(200 ) NOT NULL
```

```
)
/

-- Add keys for table Typy_zgloszen

ALTER TABLE Typy_zgloszen ADD CONSTRAINT PK_Typy_zgloszen PRIMARY KEY (Nr_typu)
/

ALTER TABLE Typy_zgloszen ADD CONSTRAINT Typ UNIQUE (Typ)
/

-- Table and Columns comments section

COMMENT ON COLUMN Typy_zgloszen.Nr_typu IS 'Numer identyfikujący typ zgłoszenia'
/
COMMENT ON COLUMN Typy_zgloszen.Typ IS 'Typ zgłoszenia'
/
COMMENT ON COLUMN Typy_zgloszen.Opis IS 'Opis typu zgłoszenia'
/

-- Table Wykonawcy

CREATE TABLE Wykonawcy(
  Nr_wykonawcy Integer NOT NULL,
  Nazwa Varchar2(200 ) NOT NULL,
  Opis Varchar2(200 ),
  Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Wykonawcy

CREATE INDEX IX_Wykonawca_ma_adres ON Wykonawcy (Nr_adresu)
/

-- Add keys for table Wykonawcy

ALTER TABLE Wykonawcy ADD CONSTRAINT PK_Wykonawcy PRIMARY KEY (Nr_wykonawcy)
/

-- Table and Columns comments section

COMMENT ON COLUMN Wykonawcy.Nazwa IS 'Nazwa wykonawcy'
/
COMMENT ON COLUMN Wykonawcy.Opis IS 'Opis wykonawcy'
/


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

ALTER TABLE Budynki ADD CONSTRAINT Zarzadza_budynkiem FOREIGN KEY (Nr_spoldzielni) REFERENCES
  Spoldzielnie (Nr_spoldzielni)
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Zatrudnia FOREIGN KEY (Nr_spoldzielni) REFERENCES
  Spoldzielnie (Nr_spoldzielni)
/



ALTER TABLE Oplaty ADD CONSTRAINT ma_oplate FOREIGN KEY (Nr_lokalu) REFERENCES
  Lokale (Nr_lokalu)
/



ALTER TABLE Lokale ADD CONSTRAINT Budynek_posiada_lokal FOREIGN KEY (Nr_budynku) REFERENCES
  Budynki (Nr_budynku)
/



ALTER TABLE Lokale ADD CONSTRAINT Zarzadza_lokalem FOREIGN KEY (Nr_spoldzielni) REFERENCES
  Spoldzielnie (Nr_spoldzielni)
/



ALTER TABLE Zgloszenia ADD CONSTRAINT Obsluguje FOREIGN KEY (Nr_spoldzielni) REFERENCES
  Spoldzielnie (Nr_spoldzielni)
/



ALTER TABLE Spoldzielnie ADD CONSTRAINT Spoldzielnia_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 Budynki ADD CONSTRAINT Budynek_ma_adres FOREIGN KEY (Nr_adresu) REFERENCES
  Adresy (Nr_adresu)
/



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



ALTER TABLE Pracownicy ADD CONSTRAINT ma_stanowisko FOREIGN KEY (Nr_stanowiska) REFERENCES
  Stanowiska (Nr_stanowiska)
```

```
/


ALTER TABLE Budynki ADD CONSTRAINT Budynek_ma_rodzaj FOREIGN KEY (Nr_rodzaju) REFERENCES
  Rodzaje_budynkow (Nr_rodzaju)
/


ALTER TABLE Zgloszenia ADD CONSTRAINT Zgloszenie_ma_typ FOREIGN KEY (Nr_typu) REFERENCES
  Typy_zgloszen (Nr_typu)
/


ALTER TABLE Budynki ADD CONSTRAINT ma_wykonawce FOREIGN KEY (Nr_wykonawcy) REFERENCES
  Wykonawcy (Nr_wykonawcy)
/


ALTER TABLE Wykonawcy ADD CONSTRAINT Wykonawca_ma_adres FOREIGN KEY (Nr_adresu) REFERENCES
  Adresy (Nr_adresu)
/
```

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

5.4.1. Wypełnienie bazy danych

```
-- Adresy
INSERT INTO Adresy VALUES(1,'Ciszewskiego','16',NULL,'Warszawa');
INSERT INTO Adresy VALUES(2,'KEN','84','15','Warszawa');
INSERT INTO Adresy VALUES(3,'Puławska','112','76','Warszawa');
INSERT INTO Adresy VALUES(4,'Ciszewskiego','16',NULL,'Warszawa');
INSERT INTO Adresy VALUES(5,'Świętokrzyska','46',NULL,'Warszawa');
INSERT INTO Adresy VALUES(6,'Marszałkowska','67',NULL,'Warszawa');
INSERT INTO Adresy VALUES(7,'Złota','44',NULL,'Warszawa');
INSERT INTO Adresy VALUES(8,'Indiri Ghandi','12','26','Warszawa');
INSERT INTO Adresy VALUES(9,'Indiri Ghandi','12','76','Warszawa');

-- Spoldzielnie
INSERT INTO Spoldzielnie VALUES(1,'Pod Skarpą','1983-06-12',1);
INSERT INTO Spoldzielnie VALUES(2,'Na Skraju','2002-10-24', 2);

-- Stanowiska
INSERT INTO Stanowiska VALUES(1,'Księgowa','Zajmuje się księgowością');
INSERT INTO Stanowiska VALUES(2,'Dozorca',
    'Zapewnienia porządek i czystość na terenie danej nieruchomości');
INSERT INTO Stanowiska VALUES(3,'Sekretarka','Zajmuje się pomocą organizacyjną w pracy prezesa');
INSERT INTO Stanowiska VALUES(4,'Pracownik','Pracuje w spółdzielni');
```

```
-- Pracownicy
INSERT INTO Pracownicy VALUES(1,'Karol','Kowalski','1973-05-26','73052611379','M','2010-06-15',
    '00876497375925047327895340','7000','jan.kowalski@gmail.com','765345098',1,3,4);
INSERT INTO Pracownicy VALUES(2,'Rico','Benitez','1985-07-30',NULL,'M','2019-10-01',
    '59438904839589385305554353','3010','rico.benitez@gmail.com','+54654893265',1,3,1);
INSERT INTO Pracownicy VALUES(3,'Jan','Waldemar','1949-02-15','49021545637','M','2002-10-24',
    '43279847983792374728942379','0','waldemar1949@o2.pl','+48800100100',2,2,2);
INSERT INTO Pracownicy VALUES(4,'Genowefa','Wiśniewska','1991-07-06','91070644461','K','2017-07-24',
    NULL,'1000',NULL,NULL,2,1,3);

-- Czlonkowie_zarzadu
INSERT INTO Czlonkowie_zarzadu VALUES(1,'2019-12-01','2024-12-01','Prezes');

-- Wykonawcy
INSERT INTO Wykonawcy VALUES(1, 'Budimex',
    'Specjalizuje się w realizacji konstrukcji żelbetowych i murowych', 2);
INSERT INTO Wykonawcy VALUES(2, 'PoliBudex',
    'Specjalizuje się w budowie obiektów mieszkalnych', 3) ;

-- Rodzaje_budynkow
INSERT INTO Rodzaje_budynkow VALUES(1, 'Mały blok',
    'Budynek mieszkalny, który ma co najwyżej 4 piętra');
INSERT INTO Rodzaje_budynkow VALUES(2, 'Blok',
    'Budynek mieszkalny, który ma co najmniej 4 piętra i co najwyżej 10');
INSERT INTO Rodzaje_budynkow VALUES(3, 'Mały dom jednorodzinny',
    'Dom z jednym lokalem i ogródkiem do 40 metrów kwadratowych ') ;

-- Typy_zgloszen
INSERT INTO Typy_zgloszen VALUES(1, 'Lokalna awaria kanalizacji',
    'Awaria kanalizacji występująca jedynie w obrębie jednego lokalu');
INSERT INTO Typy_zgloszen VALUES(2, 'Awaria windy',
    'Awaria windy w danym bloku mieszkalnym');
INSERT INTO Typy_zgloszen VALUES(3, 'Awaria kanalizacji',
    'Awaria kanalizacji występująca w obrębie wielu lokalów');

-- Budynki
INSERT INTO Budynki VALUES(1, 300, 9, 100, 10, 1, 6, 2, 2);
INSERT INTO Budynki VALUES(2, 100, NULL, 1,  NULL, 2, 5, 3, 2);
INSERT INTO Budynki VALUES(3, 250, 4, 50,  5, 1, 7, 1, 1);

-- Lokale
INSERT INTO Lokale VALUES(1, 80, NULL, 100, 5, 0, 2, 2);
INSERT INTO Lokale VALUES(2, 15, 2000, 40, 3, 3, 1, 1);
INSERT INTO Lokale VALUES(3, 16, 2000, 40, 3, 3, 1, 1);
INSERT INTO Lokale VALUES(4, 21, 3000, 60, 5, 4, 3, 1);

-- Zgloszenia
INSERT INTO Zgloszenia VALUES(1, '2022-12-05', NULL,
    'Zalanie mieszkania', NULL, 1, 1);
INSERT INTO Zgloszenia VALUES(2, '2022-12-01', '2022-12-02',
    'Winda zablokowana na piętrze 2.', 500 , 2, 2);

-- Powiazanie zgloszen z budynkami
INSERT INTO Zgloszenia_Budynki VALUES(2, 1);

-- Powiazanie zgloszen z lokalami
```

```
INSERT INTO Zgloszenia_Lokale VALUES(1, 4);

-- Oplaty
INSERT INTO Oplaty VALUES(1,'2022-03-10',953,NULL,'Oplacono',1);
INSERT INTO Oplaty VALUES(2,'2022-03-10',2456,NULL,'Oplacono',2);
INSERT INTO Oplaty VALUES(3,'2022-05-10',1053,NULL,'Nieoplacono',1);
INSERT INTO Oplaty VALUES(4,'2022-05-10',2556,NULL,'Nieoplacono',2);

-- Wlasciciele
INSERT INTO Wlasciciele VALUES(1,'Milosz','Kowalski','1999-01-12','99011224114','M',
    NULL,'335345098', 5);
INSERT INTO Wlasciciele VALUES(2,'Jakub','Kowalewicz','2000-07-13', '00271365419','M',
    'wesole.pieski@gmail.com',NULL,6);

-- Powiazanie wlascicieli z lokalami
INSERT INTO Wlasciciele_Lokale VALUES(1,1);
INSERT INTO Wlasciciele_Lokale VALUES(2,2);
INSERT INTO Wlasciciele_Lokale VALUES(2,4);

-- Lokatorzy
INSERT INTO Lokatorzy VALUES(1,'Milosz','Kowalski','1999-01-12','99011224114','M',NULL,'335345098');
INSERT INTO Lokatorzy VALUES(2,'Jakub', 'Bąk','1987-02-14',NULL,'M','jakub.bak@gmail.com',NULL);
INSERT INTO Lokatorzy VALUES(3,'Aneta', 'Nowakowska','1989-11-18',NULL,'K',NULL,'600765400');

-- Powiazanie lokatorow z lokalami
INSERT INTO Lokatorzy_Lokale VALUES(1,1);
INSERT INTO Lokatorzy_Lokale VALUES(2,4);
INSERT INTO Lokatorzy_Lokale VALUES(3,4);
```

5.4.2. Przykłady zapytań i poleceń SQL

Znajdź imiona, nazwiska oraz adresy e-mail pracowników, którzy pracują na stanowisku **Księgowa** w spółdzielni **Pod Skarpą**:

```
SELECT p.imie, p.nazwisko, p.email
FROM Spoldzielnie s , Pracownicy p , Stanowiska st
    WHERE s.nr_spoldzielni = p.nr_spoldzielni
        AND p.nr_stanowiska = st.nr_stanowiska
        AND st.nazwa_stanowiska = 'Księgowa'
        AND s.nazwa = 'Pod Skarpą';
```

Listing 1: Zapytanie

IMIE	NAZWISKO	EMAIL
Rico	Benitez	rico.benitez@gmail.com

Rysunek 1: Wynik zapytania

Znajdź imiona, nazwiska, pesele oraz adresy zamieszkania właścicieli lokali, którzy mają jakieś **Nieopłacone** opłaty:

```
SELECT w.imie, w.nazwisko, w.pesel, a.ulica, a.nr_budynku, a.nr_lokalu, a.miasto
FROM Wlasciciele w INNER JOIN Adresy a ON w.nr_adresu = a.nr_adresu
WHERE w.nr_wlasciciela IN (
        SELECT wl.nr_wlasciciela
        FROM Wlasciciele_Lokale wl, Lokale, Oplaty
        WHERE wl.nr_lokalu = Lokale.nr_lokalu
        AND Lokale.nr_lokalu = Oplaty.nr_lokalu
        AND Oplaty.Status = 'Nieoplacono'
    );
```

IMIE	NAZWISKO	PESEL	ULICA	NR_BUDYNKU	NR_LOKALU	MIASTO
Milosz	Kowalski	99011224114	Świętokrzyska	46		Warszawa
Jakub	Kowalewicz	00271365419	Marszałkowska	67		Warszawa

Rysunek 2: Wynik zapytania

Listing 2: Zapytanie

Znajdź powierzchnię, liczbę kondygnacji oraz liczbę lokali budynków, które posiadają więcej niż jeden lokal:

```
SELECT Powierzchnia, Liczba_kondygnacji, Liczba_lokali
FROM Budynki
WHERE Liczba_lokali > 1;
```

POWIERZCHNIA	LICZBA_KONDYGNACJI	LICZBA_LOKALI
300	9	100
250	4	50

Rysunek 3: Wynik zapytania

Listing 3: Zapytanie

Znajdź nazwę oraz opis wykonawcy, który wykonywał budynek o **powierzchni** większej niż 100:

```
SELECT w.nazwa, w.opis
FROM Wykonawcy w INNER JOIN Budynki b
ON w.nr_wykonawcy = b.nr_wykonawcy
WHERE w.nr_wykonawcy IS NOT NULL
AND  b.powierzchnia > 100;
```

NAZWA	OPIS
Budimex	Specjalizuje się w realizacji konstrukcji żelbetowych i murowych
PoliBudex	Specjalizuje się w budowie obiektów mieszkalnych

Rysunek 4: Wynik zapytania

Listing 4: Zapytanie

Znajdź dane lokatorów, których dotyczyło zgłoszenie z dnia 5 Grudnia 2022:

```
SELECT * FROM Lokatorzy WHERE nr_lokatora IN (
    SELECT ll.nr_lokatora
    FROM Lokatorzy_Lokale ll INNER JOIN Lokale l
    ON ll.nr_lokalu = l.nr_lokalu
    INNER JOIN Zgloszenia_Lokale zl
    ON l.nr_lokalu = zl.nr_lokalu
    INNER JOIN Zgloszenia z
    ON zl.nr_zgloszenia = z.nr_zgloszenia
    WHERE z.data_zgloszenia = '2022-12-05'
);
```

NR_LOKATORA	IMIE	NAZWISKO	DATA_URODZENIA	PESEL	PLEC	EMAIL	NR_TELEFONU
2	Jakub	Bąk	14/02/1987		M	jakub.bak@gmail.com	
3	Aneta	Nowakowska	18/11/1989		K		600765400

Rysunek 5: Wynik zapytania

Listing 5: Zapytanie

Znajdź cenę najmu i powierzchnię 2 lokali z największą ceną najmu:

```
SELECT Cena_najmu, Powierzchnia
FROM Lokale
WHERE Cena_najmu IS NOT NULL
ORDER BY Cena_najmu DESC
FETCH FIRST 2 ROWS ONLY;
```

CENA_NAJ...	POWIERZCHNIA
3000	60
2000	40

Rysunek 6: Wynik zapytania

Listing 6: Zapytanie