

Projekt BDBT

-

”Operator sieci komórkowej”

-

Zespół B12

Dana Betsina

Emilia Anczarska

Politechnika Warszawska, Instytut Telekomunikacji

7 stycznia 2024

Spis treści

| | |
|---|----|
| 1. Zakres i cel projektu (opis założeń funkcjonalnych projektowanej bazy danych) | 3 |
| 2. Definicja systemu | 3 |
| 2.1. Perspektywy użytkowników | 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 | 5 |
| 3.3. Określenie atrybutów i ich dziedzin | 6 |
| 3.4. Dodatkowe reguły integralnościowe (reguły biznesowe) | 8 |
| 3.5. Klucze kandydujące i główne (decyzje projektowe) | 9 |
| 3.6. Schemat ER na poziomie konceptualnym | 10 |
| 3.7. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady | 11 |
| 3.7.1. Pułapka szczelinowa | 11 |
| 3.7.2. Pułapka wachlarzowa | 11 |
| 4. Model logiczny | 12 |
| 4.1. Charakterystyka modelu relacyjnego | 12 |
| 4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym | 12 |
| 4.2.1. Usunięcie pól segmentowych | 12 |
| 4.2.2. Usunięcie pól wielowartościowych | 13 |
| 4.2.3. Usunięcie związków N:M | 13 |
| 4.3. Proces normalizacji | 14 |
| 4.3.1. Pierwsza postać normalna | 14 |
| 4.3.2. Druga postać normalna | 14 |
| 4.3.3. Trzecia postać normalna | 15 |
| 4.4. Schemat ER na poziomie modelu logicznego | 16 |
| 4.5. Więzy integralności | 17 |
| 4.6. Proces denormalizacji – analiza i przykłady | 17 |
| 5. Faza fizyczna | 18 |
| 5.1. Projekt transakcji i weryfikacja ich wykonalności | 18 |
| 5.2. Strojenie bazy danych – dobór indeksów | 18 |
| 5.3. Skrypt SQL zakładający bazę danych | 20 |
| 5.4. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych | 33 |
| 5.4.1. Podgląd ofert tańszych niż 40 złotych | 33 |
| 5.4.2. Podgląd planów komórkowych z limitem internetu większym lub równym 20Gb | 33 |
| 5.4.3. Podgląd umów za październik 2023 | 33 |
| 5.4.4. Podgląd klientów, którzy mieszkają w Białymstoku | 34 |
| 5.4.5. Podgląd serwisantów posiadających prawo jazdy | 34 |
| 5.4.6. Podgląd pracownika, który podpisał najwięcej umów w placówce o id_placówki = 7 | 34 |
| 6. Bibliografia | 34 |

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

Celem projektu było stworzenie bazy danych dla **operatora sieci komórkowej**. Kolejne encje reprezentują zatrudnianych przez operatora **pracowników**, posiadanych przez niego **klientów** oraz proponowane **oferty**, które mogą zawierać urządzenia, takie jak **modemy**, **routery** i **telefony**. Dodatkowo uwzględniamy **placówkę** operatora i **umowę** jaka operator podpisuje z klientem.

Ponieważ nie dysponowaliśmy wiedzą ekspercką dotyczącą opisywanego zjawiska, zdane na własną intuicję i wobrażenie na temat działalności takiego operatora ustaliliśmy pewne "reguły biznesowe". Wszystkie te decyzje znajdują odzwierciedlenie w fazie konceptualnej oraz logicznej naszego projektu i zostały dokładnie opisane w dalszych rozdziałach.

Funkcjonalność bazy powinna obejmować możliwość modyfikowania, czyli dodawania i usuwania klientów oraz pracowników z bazy danych.

2. Definicja systemu

Zakładamy, że nasz system posiada następujące funkcjonalności:

1. dostęp do informacji o:
 - a) operatorze sieci komórkowej
 - b) właścicielu
 - c) posiadanych placówkach
 - d) zatrudnionych pracownikach
 - e) ich stanowiskach oraz specjalnościach
 - f) ofertach operatora
 - g) zawartych umowach z klientami
 - h) planach komórkowych
 - i) oferowanych urządzeniach
 - j) klientach
 - k) zarejestrowanych kartach SIM w sieci operatora
2. możliwość dodawania/modyfikowania/usuwania informacji o wyżej wymienionych podmiotach

2.1. Perspektywy użytkowników

Użytkownicy tej bazy to klient, pracownik placówki, manager działu marketingu, administrator systemu, manager, właściciel.

1. Administrator systemu - dostęp i możliwość modyfikacji wszystkich danych w bazie
2. Właściciel - dostęp do wszystkich danych w bazie i podglądu
3. Manager - dostęp do danych związanych z pracownikami i umowami w bazie oraz do ich podglądu
4. Klient - dostęp do danych o własnych, zawartych umowach, ofertach, informacji o operatorze i dostępnych planach komórkowych i urządzeń
5. Pracownik - dostęp do informacji o zawartych umowach, ofertach i klientach w bazie

Mają oni przykładowe, ze względu na potrzeby możliwości dostępu do zasobów bazy danych:

| | Właściciel | Manager | Manager działu marketingu | Administrator systemu | Pracownik | Klienci |
|---|------------|---------|---------------------------|-----------------------|-----------|---------|
| Wyszukiwanie ofert dostępnych w danym okresie | + | - | + | + | + | - |
| Zarządzanie swoim kontem | + | + | - | - | - | - |
| Wyszukiwanie planów | + | - | - | - | - | - |
| Przegląd dostępnych placówek | + | - | - | + | + | + |
| Dostęp do danych klienckich | + | + | + | + | - | - |
| Dostęp do podpisanych umów | + | + | + | - | - | + |
| Wyszukiwanie urzędzeń | + | - | - | - | + | + |
| Dostęp do danych pracowników | + | + | - | + | + | - |
| Wyszukiwanie danych operatora | + | - | - | + | - | + |

3. Model konceptualny

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

Encje, które określiliśmy w projekcie to:

1. Operator - główna encja opisująca Operatora, którego dane znajdują się w naszej bazie
2. Placówka - encja, która zawiera informacje opisujące posiadane przez operatora placówki
3. Pracownik - encja, która zawiera informacje o osobach zatrudnionych przez operatora, podlega podziałowi na stanowiska, takie jak serwisant i handlowiec
4. Serwisant - uszczegółowienie pracownika sieci operatorskiej jako serwisanta
5. Handlowiec - uszczegółowienie pracownika sieci operatorskiej jako handlowca
6. Klient - informacje o klientach, którzy korzystają z usług operatora sieci komórkowej
7. Umowa - informacje o podpisywanym ofertach pomiędzy klientami a handlowcami
8. Oferta - informacje o proponowanych przez operatora ofertach
9. Plan Komórkowy - szczegółowe informacje dotyczące usług operatorskich możliwych do wyboru
10. Urządzenie - informacje o urządzeniach, mogących się znaleźć w ofertach operatora sieci, podlega uszczegółowieniu na typ urządzenia
11. Telefon - uszczegółowienie Urządzenia
12. Router - uszczegółowienie Urządzenia
13. Modem - uszczegółowienie Urządzenia

3.2. Ustalenie związków między encjami i ich typów

| Nazwa związku | Typ związku | Typ uczestnictwa | Opis |
|-----------------------------|----------------|---------------------------|--|
| Operator - Placówka | Jeden do wielu | obowiązkowe - opcjonalne | Operator posiada wiele placówek, placówka należy do jednego operatora |
| Operator - Oferta | Jeden do wielu | obowiązkowe - opcjonalne | Operator proponuje wiele ofert, oferta jest proponowana przez jednego operatora |
| Operator - Pracownik | Jeden do wielu | obowiązkowe - opcjonalne | Operator zatrudnia wiele pracowników, pracownik jest zatrudniany przez operatora |
| Operator - Klient | Jeden do wielu | obowiązkowe - opcjonalne | Operator posiada wiele klientów, klient korzysta z usług jednego operatora |
| Placówka - Pracownik | Jeden do wielu | obowiązkowe - obowiązkowe | Placówka podpisuje umowę o pracę z wieloma pracownikami, pracownik podpisuje umowę z jedną placówką |
| Pracownik - Umowa | Jeden do wielu | obowiązkowe - opcjonalnie | Placówka podpisuje umowę o pracę z wieloma pracownikami, pracownik podpisuje umowę z jedną placówką |
| Klient - Umowa | Jeden do wielu | opcjonalne - obowiązkowe | Klient podpisuje wiele umów bądź zero, umowa jest podpisywana przez jednego klienta |
| Placówka - Umowa | Jeden do wielu | opcjonalne - opcjonalne | W placówce podpisywane jest wiele umów bądź zero, umowa jest podpisywana w jednej placówce bądź w żadnej |
| Oferta - Umowa | Jeden do wielu | obowiązkowe - opcjonalnie | Umowa zawiera warunki oferty, na podstawie oferty zawierana jest jedna umowa |
| Plan komórkowy - Oferta | Jeden do wielu | obowiązkowe - opcjonalnie | Oferta posiada plan, plan zawarty jest w wielu ofertach |
| Plan komórkowy - Urządzenie | Wielu do wielu | opcjonalne - opcjonalne | W planie nie musi być zawarte żadne urządzenie, urządzenie może nie być zawarte w żadnym planie |

3.3. Określenie atrybutów i ich dziedzin

Operator:

| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
|----------------|-----------------|---------------|--|
| id_operatora | Integer | obowiązkowe | unikatowy numer operatora |
| nazwa | VarChar(300) | obowiązkowe | pełna nazwa operatora |
| wlasciciel | VarChar(200) | obowiązkowe | nazwa właściciela sieci operatorskiej |
| data_zalozenia | Date | obowiązkowe | dokładna data rozpoczęcia działalności operatora |
| adres_biura | VarChar(100) | obowiązkowe | adres głównego biura sieci operatorskiej |

Placówka:

| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
|----------------|-----------------|---------------|---|
| id_placowki | Integer | obowiązkowe | unikatowy numer placówki podlegającej operatorowi |
| adres | VarChar(100) | Obowiązkowe | adres placówki |
| telefon | VarChar(12) | obowiązkowe | numer kontaktowy do biura placówki |
| adres_email | VarChar(50) | obowiązkowe | adres kontaktowy do biura placówki |

Pracownik:

| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
|--------------------|-----------------|----------------|--|
| id_pracownika | Integer | obowiązkowe | unikatowy numer pracownika zatrudnionego przez placówkę |
| imie | VarChar(20) | obowiązkowe | imię pracownika |
| nazwisko | VarChar(30) | obowiązkowe | nazwisko pracownika |
| pesel | VarChar(11) | nieobowiązkowe | numer pesel pracownika |
| plec | Domain(plecD) | obowiązkowe | pleć pracownika |
| stanowisko | VarChar(50) | obowiązkowe | stanowisko zajmowane przez pracownika sieci |
| numer_telefonu | VarChar(12) | obowiązkowe | numer kontaktowy do pracownika |
| wynagrodzenie | Money | nieobowiązkowe | otrzymywane przez pracownika wynagrodzenie |
| adres_zamieszkania | VarChar(100) | obowiązkowe | dokładny adres pracownika. Zawiera: miejscowość, kod pocztowy, ulica, numer budynku, ewentualnie nr mieszkania |
| numer_konta | VarChar(24) | nieobowiązkowe | wynagrodzenie otrzymywane przez pracownika |
| numer_placowki | Integer | obowiązkowe | numer placówki, w której pracownik jest zatrudniony |
| data_urodzenia | Date | obowiązkowe | data urodzenia pracownika |
| adres_email | VarChar(50) | obowiązkowe | adres mailowy pracownika |

Umowa:

| Nazwa atrybutu | Typ i dziedzina | Obowiązkowość | Opis |
|-----------------|-----------------|---------------|-------------------------------------|
| id_umowy | Integer | obowiązkowe | unikatowy numer umowy |
| data_podpisania | Date | obowiązkowe | data podpisania umowy przez klienta |
| karta_sim | VarChar(200) | obowiązkowe | numer karty SIM klienta |

Oferta:

| Nazwa atrybuty | Typ i dziedzina | Obowiązkowość | Opis |
|----------------|-----------------|---------------|---|
| id_oferty | Integer | obowiązkowe | unikatowy numer oferty |
| nazwa | VarChar(300) | obowiązkowe | pełna nazwa oferty |
| okres | VarChar(30) | obowiązkowe | określa długość trwania oferty w miesiącach |
| cena | Money | obowiązkowe | cena oferty |

Plan komórkowy:

| Nazwa atrybuty | Typ i dziedzina | Obowiązkowość | Opis |
|--------------------|-----------------|----------------|--|
| id_planu | Integer | obowiązkowe | unikatowy numer planu |
| limit_internetu | Integer | obowiązkowe | limit internetu do wykorzystania |
| limit_internetu_EU | Integer | obowiązkowe | limit internetu poza granicami kraju w strefie EU do wykorzystania |
| limit_sms | Integer | obowiązkowe | limit możliwych SMS-ów do wysłania |
| limit_mms | Integer | obowiązkowe | limit możliwych MMS-ów do wysłania |
| roaming | Integer | nieobowiązkowe | limit internetu do wykorzystania poza granicami kraju |

Urządzenie:

| Nazwa atrybuty | Typ i dziedzina | Obowiązkowość | Opis |
|----------------|-----------------|---------------|----------------------------|
| id_urzadzenia | Integer | obowiązkowe | unikatowy numer urządzenia |
| model | VarChar(30) | obowiązkowe | nazwa modelu urządzenia |
| marka | VarChar(30) | obowiązkowe | nazwa marki urządzenia |
| cena | Money | obowiązkowe | cena urządzenia |
| nazwa | VarChar(50) | obowiązkowe | |

Klient:

| Nazwa atrybuty | Typ i dziedzina | Obowiązkowość | Opis |
|--------------------|-----------------|----------------|--|
| id_klienta | Integer | obowiązkowe | unikatowy numer klienta korzystającego z usług |
| imie | VarChar(20) | obowiązkowe | imię klienta |
| nazwisko | VarChar(30) | obowiązkowe | nazwisko klienta |
| pesel | VarChar(11) | nieobowiązkowe | numer pesel klienta |
| numer_telefonu | VarChar(12) | obowiązkowe | numer telefonu klienta |
| adres_zamieszkania | VarChar(100) | obowiązkowe | dokładny adres klienta. Zawiera: miejscowość, kod pocztowy, ulica, numer budynku |

Serwisant:

| Nazwa atrybuty | Typ i dziedzina | Obowiązkowość | Opis |
|----------------|-----------------|---------------|---|
| prawo_jazdy | Boolean | obowiązkowe | wartość wskazująca czy serwisant ma prawo jazdy |
| zawod | VarChar(300) | obowiązkowe | opis zawodu wykonywanego |

Handlowiec:

| Nazwa atrybuty | Typ i dziedzina | Obowiązkowość | Opis |
|-------------------|-----------------|---------------|---|
| znajomosc_jezykow | VarChar(300) | obowiązkowe | opis certyfikatów i znajomości języków handlowca |
| szkolenie_kasowe | Boolean | obowiązkowe | wartość wskazująca czy handlowiec odbył szkolenie |

Telefon:

| Nazwa atrybuty | Typ i dziedzina | Obowiązkowość | Opis |
|----------------|-----------------------------|---------------|---|
| typ_karty | typKarty(Micro, Mini, Nano) | obowiązkowe | typ karty używanej w telefonie |
| pamiec | VarChar(30) | obowiązkowe | pamięć telefonu |
| eSIM | Boolean | obowiązkowe | wartość wskazująca czy telefon posiada dodatkowo kartę eSIM |

Router:

| Nazwa atrybuty | Typ i dziedzina | Obowiązkowość | Opis |
|----------------|-------------------------|---------------|--------------------------------|
| liczba_anten | liczbaAnten(0, 2, 4, 8) | obowiązkowe | liczba anten w routerze |
| predkosc | VarChar(15) | obowiązkowe | maksymalna prędkość połączenia |
| czestotliwosc | VarChar(30) | obowiązkowe | częstotliwości odbieranych fal |

Modem:

| Nazwa atrybuty | Typ i dziedzina | Obowiązkowość | Opis |
|----------------|--------------------------------|---------------|--------------------------------|
| przeznaczenie | przeznaczenie(3G, LTE, 4G, 5G) | obowiązkowe | opis technologii |
| interfejs | interfejs(USB-C, USB-2.0) | obowiązkowe | typ wtyczki |
| czestotliwosc | VarChar(15) | obowiązkowe | częstotliwości odbieranych fal |

3.4. Dodatkowe reguły integralnościowe (reguły biznesowe)

W ramach dodatkowych reguł integralnościowe założonych na potrzeby spójności projektu, zakładamy, że:

- Urządzenia, które są zawarte w umowie, nie są otrzymane przez klienta bezpośrednio po podpisaniu umowy w placówce oraz nie są prezentowane w placówkach operatora. Są dostarczane na adres klienta, bądź inny wskazany adres.
- Dane klientów, pracowników, podpisywanych ofert i umów spełniają określone kryteria:
 - numer konta bankowego składa się z 24 cyfr.
 - numer telefonu jest unikatowy w całej bazie danych i posiada następującą strukturę: XXX-XXX-XXX
 - pesel pracowników oraz klientów jest unikatowy w całej bazie danych i składa się z 11 cyfr o konkretnie określonej strukturze.
 - okres trwania oferty jest zawsze wyrażony w miesiącach
- Relacje encji w bazie danych nie muszą być obowiązkowo spełnione:
 - operator może nie posiadać żadnych ofert, placówek, klientów, pracowników
 - placówka może nie posiadać żadnych umów
 - pracownik może nie podpisać żadnych umów
 - oferta może nie być zawarta w żadnej umowie
 - plan komórkowy może nie być zawarty w żadnej ofercie i może nie zawierać żadnych urządzeń
 - urządzenie może nie być zawarte w żadnym planie

3.5. Klucze kandydujące i główne (decyzje projektowe)

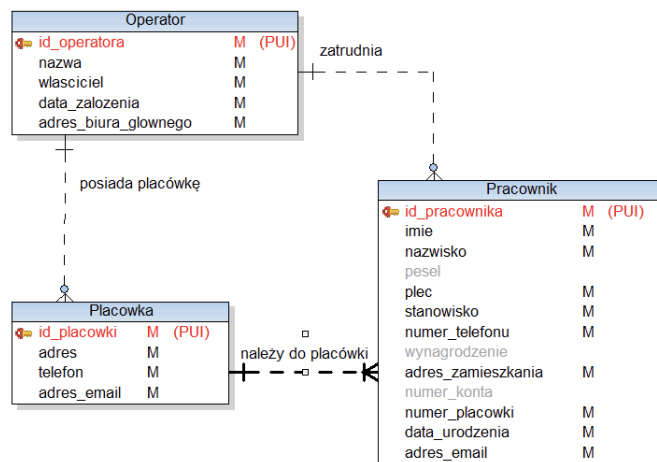
| Encja | Klucz główny | Klucz kandydujący |
|----------------|---------------|-------------------|
| Operator | id_operatora | nazwa |
| Placowka | id_placowki | adres |
| Pracownik | id_pracownika | pesel |
| Oferta | id_oferty | nazwa |
| Klient | id_klienta | numer_telefonu |
| Umowa | id_umowy | - |
| Plan komórkowy | id_planu | - |
| Urządzenie | id_urzadzenia | - |

10



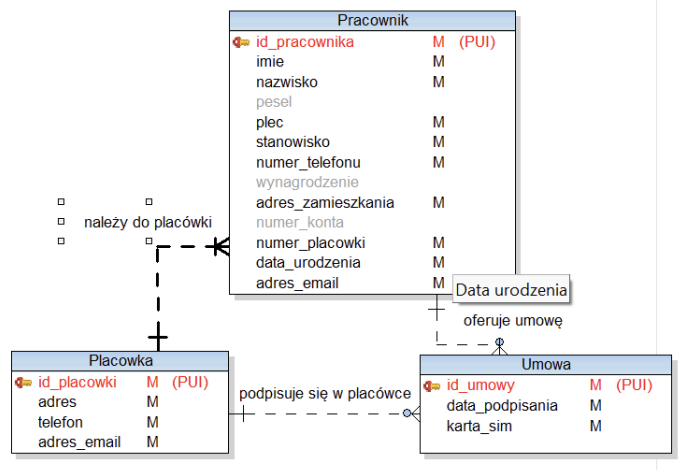
3.7. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

3.7.1. Pułapka szczelinowa



W pierwszych wersjach naszego projektu zdecydowaliśmy, że każdy pracownik musi być zatrudniony w którejś z placówek operatora sieci komórkowej. Jeżeli jednak chcielibyśmy wprowadzić pracowników, którzy nie są przypisani do żadnej z nich, np. administrator systemu, powstałaby pułapka szczelinowa. Aby rozwiązać ten problem dodałyśmy relację Operator - Pracownik.

3.7.2. Pułapka wachlarzowa



Umowa musi być podpisana przez jednego z pracowników zatrudnionego przez naszego Operatora. Jeżeli jednak klient chciałby znać dane tego pracownika, nie mógłby tego znaleźć poprzez atrybuty umowy i utworzyłaby się pułapka wachlarzowa. Aby tego uniknąć dodałyśmy relację Pracownik - Umowa.

4. Model logiczny

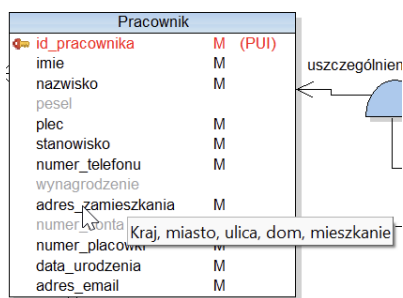
4.1. Charakterystyka modelu relacyjnego

Po zakończeniu prac nad modelem konceptualnym i weryfikacji jego poprawności dokonaliśmy procesu konwersji projektu do poziomu logicznego.

1. Relacja ma nazwę, która jest odmienna od wszystkich innych nazw pozostałych relacji
2. Każdy atrybut w ramach danej relacji ma inną nazwę
3. Wszystkie wartości atrybutu pochodzą z tej samej dziedziny
4. Nazwy tabel zostały zmienione z liczby pojedynczej na mnogą
5. Identyfikujący atrybut każdej encji stał się kluczem głównym tabeli
6. Każda relacja "wiele do wielu" została zastąpiona dwiema relacjami "jeden do wielu" wraz z tabelą łączącą obie nowo powstałe relacje
7. Tabele otrzymały dodatkowe atrybuty w postaci kluczy obcych (zaznaczone na zielono)

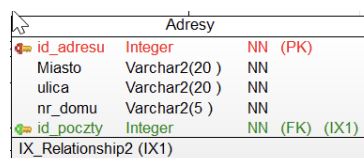
4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym

Tworząc model logiczny trzeba było przeprowadzić kilka operacji w celu uzyskania kompatybilnego modelu. Na przykładzie encji *Pracownik* z modelu konceptualnego pokażemy dokonane zmiany.



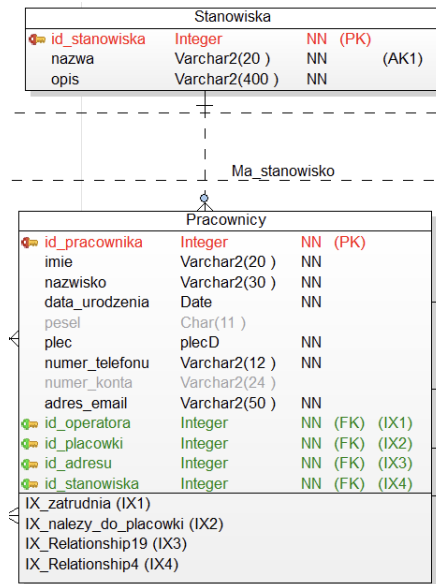
4.2.1. Usunięcie pól segmentowych

Jak widać na obrazku powyżej, atrybut *adres_zamieszkania* składa się z kilku różnych typów wartości: Miasto, numer domu itd. W celu rozwiązania tego problemu stworzyliśmy osobną encję *Adresy*, która zawiera w sobie kilka atrybutów.



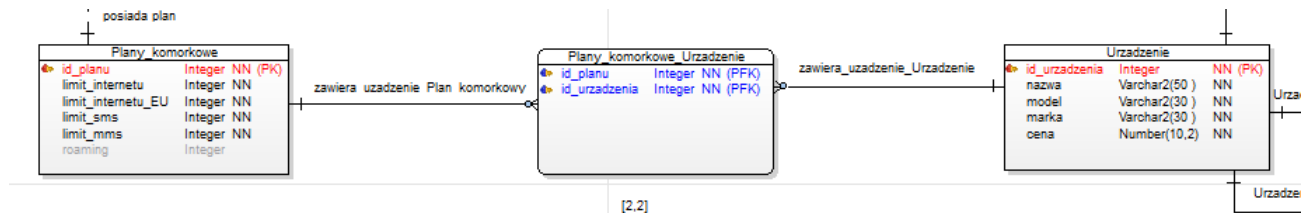
4.2.2. Usunięcie pól wielowartościowych

Atrybut *stanowisko* w modelu konceptualnym musi być przykonwertowany do słownika wartości w modelu logicznym, ponieważ atrybut nie może się składać z kilku instancji tego samego typu wartości.



4.2.3. Usunięcie związków N:M

Tabele **Plan_komorkowy** i **Urządzenie** w modelu konceptualnym łączyła relacja typu "wielu do wielu". Przy przejściu na model logiczny została utworzona tabela **Plany_komorkowe_Urządzenie**, której zadaniem jest przechowywać informacje o tym, które urządzenie jest zawarte w konkretnym planie.



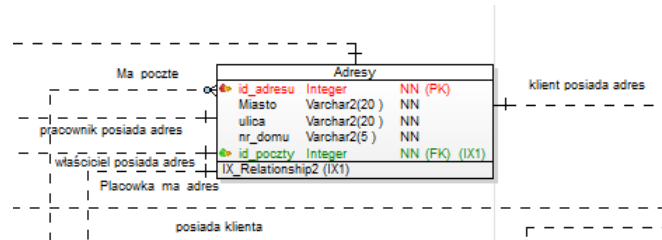
4.3. Proces normalizacji

Celem procesu normalizacji bazy danych jest eliminacja powtarzających się danych w bazie oraz trzymaniu ich w jednym miejscu.

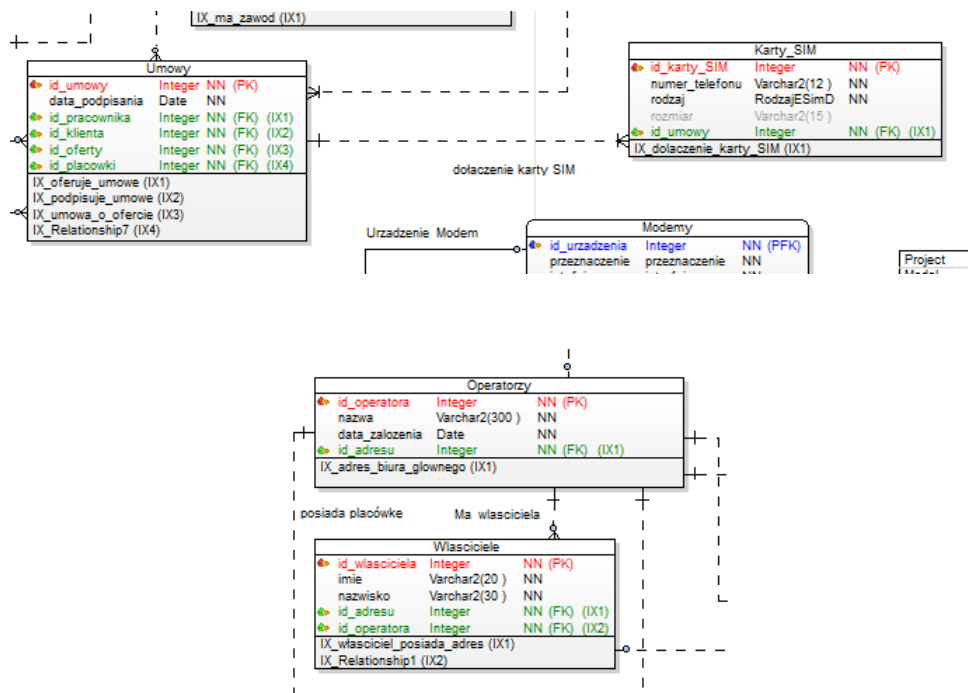
4.3.1. Pierwsza postać normalna

Pierwsza postać normalna dotyczy atomowości danych. Każdy wiersz przechowuje informacje o pojedynczym obiekcie, posiada klucz główny. Aby uzyskać tę postać w naszej bazie danych dokonaliśmy następujących zmian:

Na etapie usunięcia właściwości niekompatybilnych z modelem relacyjnym z tabel **Operatorzy**, **Pracownicy**, **Właściciele**, **Klienci** oraz **Placówki** usunęliśmy atrybut adres, ponieważ doprowadziłoby to do powstania pola segmentowego. Dodaliśmy dodatkową tabelę **Adresy**, aby przechowywać elementy adresu osobno.



Analogicznie z tabeli **Umowy** usunęliśmy atrybut *karta_SIM*, dodając nową tabelę z informacjami dotyczącymi karty, z tabeli **Operatorzy** wyodrębniliśmy tabelę **Właściciele**, z szczegółowymi informacjami o każdym właścicielu.



4.3.2. Druga postać normalna

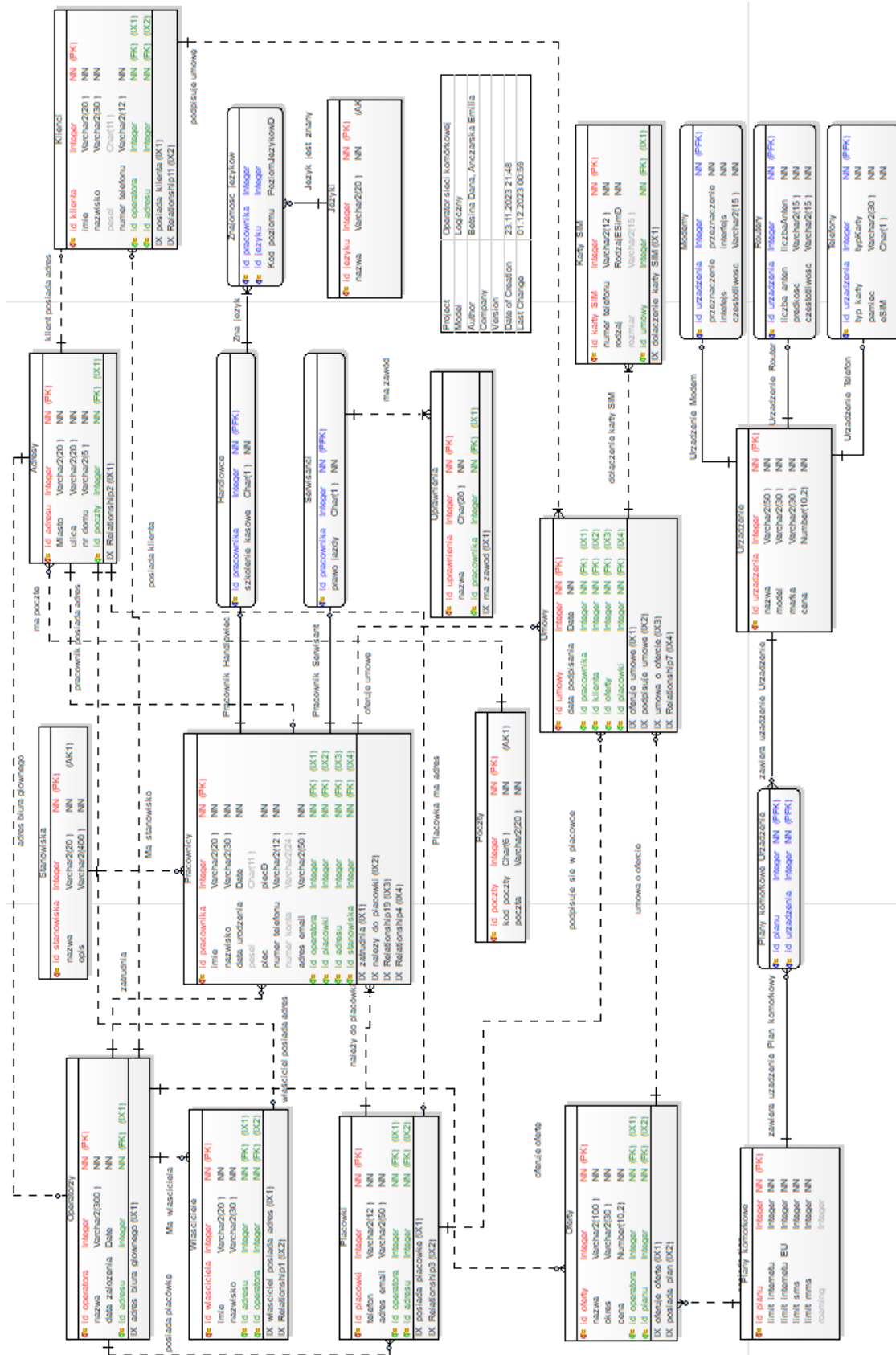
Po procesie osiągnięcia pierwszej postaci normalnej, aby uzyskać jej drugą postać musieliśmy zadbać o to, aby każdy atrybut, który nie wchodzi w skład żadnego klucza potencjalnego jest w pełni funkcyjnie zależny od wszystkich kluczy potencjalnych tej relacji. Dodatkowo każdy atrybut relacji nie wchodzący w skład klucza musi zależeć od całego klucza, a nie tylko jego części.

Ponieważ w naszej bazie każdy klucz główny jest określony poprzez identyfikator ID oraz wszystkie atrybuty informacyjne, nie należące do klucza, zawierają informacje o elementach konkretnej encji, kryterium drugiej postaci normalnej zostało spełnione.

4.3.3. Trzecia postać normalna

Po procesie weryfikacji naszego projektu z kryteriami pierwszej i drugiej postaci normalnej, przeszliśmy do wymogów trzeciej. Każdy atrybut danej relacji nie wchodzący w skład żadnego klucza potencjalnego nie może być przechodnio funkcyjnie zależny od żadnego klucza potencjalnego tej relacji, a więc był określony tylko i wyłącznie kluczem. Oznacza to, że każdy niekluczowy argument jest bezpośrednio zależny tylko od klucza głównego, a nie innej kolumny. Ponieważ wszystkie niekluczowe atrybuty w naszej bazie mogą być zidentyfikowane dzięki prostym kluczom głównym, wymogi trzeciej postaci normalnej zostały automatycznie spełnione.

16



4.5. Więzy integralności


Aby spełnić warunki integralności danych trzeba zapewnić zgodność pomiędzy zawartością pól rekordów oraz przyjmowanymi przez nie typami. Każdy klucz główny jest unikatowy i prawie każdy inny atrybut jest polem obowiązkowym, więc przypadków, gdzie któryś z nich będzie przyjmował wartość NULL jest niewiele. Każde pole jest wartością atomową.

4.6. Proces denormalizacji – analiza i przykłady

Proces denormalizacji jest procesem odwrotnym do normalizacji, aby przyspieszyć dostęp do danych kosztem wystąpienia kontrolowanej redundancji i złagodzenia pewnych reguł normalizacji.

Możemy jako przykład podać sytuację: pracownik działu marketingu potrzebuje szybkiego i łatwego dostępu do kilku atrybutów z różnych encji, które są dla niego potrzebne do przeprowadzenia analizy marketingowej. W takim przypadku tworzymy tabelę, która będzie zawierała od razu pełną potrzebną dla naszego użytkownika informację:

- numer klienta, który podpisał umowę
- numer placówki, w której została ta umowa podpisana
- numer oferty zawartej w umowie
- inne dodatkowe atrybuty

| Raport_podpisania_umowy | | |
|--|---------------|---------|
|  id_raportu | Integer | NN (PK) |
| data_raportu | Date | NN |
| id_klienta | Integer | NN |
| imie_klienta | Varchar2(20) | |
| nazwisko_klienta | Varchar2(30) | |
| id_pracownika | Integer | |
| imie_pracownika | Varchar2(20) | |
| nazwisko_pracownika | Varchar2(30) | |
| id_placówki | Integer | NN |
| id_oferty | Integer | NN |

5. Faza fizyczna

5.1. Projekt transakcji i weryfikacja ich wykonalności

Weryfikacja dostępu do informacji o:

| Funckjonalność | Weryfikacja wykonalności | Potrzebne dane |
|--|--------------------------|--|
| operatorze sieci komórkowej | + | Operatorzy, Adresy, Wlasciciele |
| właścicielu | + | Wlasciciele, Adresy, Operatorzy |
| posiadanych placówkach | + | Placowki, Operatorzy, Adresy |
| zatrudnionych pracownikach | + | Pracownicy, Adresy, Placowki, Stanowiska, Operatorzy |
| ofertach operatora | + | Oferty, Operatorzy, Plany_komorkowe |
| zawartych umowach z klientami | + | Umowy, Klienci, Placowki, Pracownicy, Karty_SIM |
| planach komórkowych | + | Plany_komorkowe, Urzadzenia |
| oferowanych urządzeniach | + | Urzadzenia |
| klientach | + | Klienci, Adresy, Umowy, Operatorzy |
| zarejestrowanych kartach SIM w sieci operatora | + | Karty_SIM, Umowy |

Co za tym idzie w bazie jesteśmy w stanie również dodawać/modyfikować/usuwać informacje o wyżej wymienionych podmiotach

5.2. Strojenie bazy danych – dobór indeksów

Do strojenia bazy danych służyć będą indeksy wygenerowane automatycznie przy okazji tworzenia relacji między tabelami.

1. IX_adres_biura_glownego - wyszukiwane adresu biura głównego operatora.

```
1 CREATE INDEX IX_adres_biura_glownego ON Operatorzy (id_adresu)
```

2. IX_zatrudnia - wyszukiwanie operatora, który zatrudnia pracowników.

```
1 CREATE INDEX IX_zatrudnia ON Pracownicy (id_operatora)
```

3. INDEX IX_nalezy_do_placowki - wyszukiwanie placówki po zatrudnionych w nich pracownikach.

```
1 CREATE INDEX IX_nalezy_do_placowki ON Pracownicy (id_placowki)
```

4. INDEX IX_oferuje_umowe - wyszukiwanie pracownika oferującego umowy.

```
1 CREATE INDEX IX_oferuje_umowe ON Umowy (id_pracownika)
```

5. INDEX IX_podpisuje_umowe - wyszukiwanie klienta, który podpisał umowę.

```
1 CREATE INDEX IX_podpisuje_umowe ON Umowy (id_klienta)
```

6. INDEX IX_umowa_o_ofercie - wyszukiwanie oferty zawartej w umowie.

```
1 CREATE INDEX IX_umowa_o_ofercie ON Umowy (id_oferty)
```

7. INDEX IX_posiada_klienta - wyszukiwanie operatora, który posiada klientów.

```
1 CREATE INDEX IX_posiada_klienta ON Klienci (id_operatora)
```

8. INDEX IX_oferuje_oferte - wyszukiwanie operatora po ofertach.

```
1 CREATE INDEX IX_oferuje_oferte ON Oferty (id_operatora)
```

9. INDEX IX_posiada_plan - wyszukiwanie planu zawartego w ofercie.

```
1 CREATE INDEX IX_posiada_plan ON Oferty (id_planu)
```

10. INDEX IX_posiada_placowke - wyszukiwanie operatora posiadającego placówki.

```
1      CREATE INDEX IX_posiada_placowke ON Placowki (id_operatora)
```

11. INDEX IX_wlasciciel_posiada_adres - wyszukiwanie adresu właściciela.

```
1      CREATE INDEX IX_wlasciciel_posiada_adres ON Wlasciciele (id_adresu)
```

12. INDEX IX_dolaczenie_karty_SIM - wyszukiwanie umowy podpisanej zawierającej kartę SIM.

```
1      CREATE INDEX IX_dolaczenie_karty_SIM ON Karty_SIM (id_umowy)
```

13. INDEX IX_ma_zawod - wyszukiwanie pracownika posiadającego uprawnienia do wykonywania zawodu.

```
1      CREATE INDEX IX_ma_zawod ON Uprawnienia (id_pracownika)
```

5.3. Skrypt SQL zakładający bazę danych

```
1  /*
2  Created: 23.11.2023
3  Modified: 10.12.2023
4  Project: Operator sieci komorkowej
5  Model: Logiczny
6  Author: Betsina Dana, Anczarska Emilia
7  Database: Oracle 19c
8  */
9
10
11 -- Create tables section -----
12
13 -- Table Operatorzy
14
15 CREATE TABLE Operatorzy(
16     id_operatora Integer GENERATED ALWAYS AS IDENTITY(
17         START WITH 1
18         INCREMENT BY 1
19         NOMAXVALUE
20         NOMINVALUE
21         CACHE 20) NOT NULL,
22     nazwa Varchar2(300 ) NOT NULL,
23     data_zalozenia Date NOT NULL,
24     id_adresu Integer NOT NULL
25 )
26 /
27
28 -- Create indexes for table Operatorzy
29
30 CREATE INDEX IX_adres_biura_glownego ON Operatorzy (id_adresu)
31 /
32
33 -- Add keys for table Operatorzy
34
35 ALTER TABLE Operatorzy ADD CONSTRAINT id_operatora_key_1 PRIMARY KEY (id_operatora)
36 /
37
38 -- Table Pracownicy
39
40 CREATE TABLE Pracownicy(
41     id_pracownika Integer GENERATED ALWAYS AS IDENTITY(
42         START WITH 1
43         INCREMENT BY 1
44         NOMAXVALUE
45         NOMINVALUE
46         CACHE 20) NOT NULL,
47     imie Varchar2(20 ) NOT NULL,
48     nazwisko Varchar2(30 ) NOT NULL,
49     data_urodzenia Date NOT NULL,
50     pesel Char(11 ),
51     plec Varchar2(1 ) NOT NULL
52         CHECK ((plec IN ('K','M'))),
53     numer_telefonu Varchar2(12 ) NOT NULL,
54     numer_konta Varchar2(24 ),
```

```

55     adres_email Varchar2(50 ) NOT NULL,
56     id_operatora Integer NOT NULL,
57     id_placowki Integer NOT NULL,
58     id_adresu Integer NOT NULL,
59     id_stanowiska Integer NOT NULL
60 )
61 /
62
63 -- Create indexes for table Pracownicy
64
65 CREATE INDEX IX_zatrudnia ON Pracownicy (id_operatora)
66 /
67
68 CREATE INDEX IX_nalezy_do_placowki ON Pracownicy (id_placowki)
69 /
70
71 CREATE INDEX IX_Relationship19 ON Pracownicy (id_adresu)
72 /
73
74 CREATE INDEX IX_Relationship4 ON Pracownicy (id_stanowiska)
75 /
76
77 -- Add keys for table Pracownicy
78
79 ALTER TABLE Pracownicy ADD CONSTRAINT id_operatora_key PRIMARY KEY (id_pracownika)
80 /
81
82 -- Table Umowy
83
84 CREATE TABLE Umowy(
85     id_umowy Integer GENERATED ALWAYS AS IDENTITY(
86         START WITH 1
87         INCREMENT BY 1
88         NOMAXVALUE
89         NOMINVALUE
90         CACHE 20) NOT NULL,
91     data_podpisania Date NOT NULL,
92     id_pracownika Integer NOT NULL,
93     id_klienta Integer NOT NULL,
94     id_oferty Integer NOT NULL,
95     id_placowki Integer NOT NULL
96 )
97 /
98
99 -- Create indexes for table Umowy
100
101 CREATE INDEX IX_oferuje_umowe ON Umowy (id_pracownika)
102 /
103
104 CREATE INDEX IX_podpisuje_umowe ON Umowy (id_klienta)
105 /
106
107 CREATE INDEX IX_umowa_o_ofercie ON Umowy (id_oferty)
108 /
109
110 CREATE INDEX IX_Relationship7 ON Umowy (id_placowki)

```

```

111 /
112
113 -- Add keys for table Umowy
114
115 ALTER TABLE Umowy ADD CONSTRAINT id_umowy_key PRIMARY KEY (id_umowy)
116 /
117
118 -- Table Klienci
119
120 CREATE TABLE Klienci(
121     id_klienta Integer GENERATED ALWAYS AS IDENTITY(
122         START WITH 1
123         INCREMENT BY 1
124         NOMAXVALUE
125         NOMINVALUE
126         CACHE 20) NOT NULL,
127     imie Varchar2(20 ) NOT NULL,
128     nazwisko Varchar2(30 ) NOT NULL,
129     pesel Char(11 ),
130     numer_telefonu Varchar2(12 ) NOT NULL,
131     id_operatora Integer NOT NULL,
132     id_adresu Integer NOT NULL
133 )
134 /
135
136 -- Create indexes for table Klienci
137
138 CREATE INDEX IX_posiada_klienta ON Klienci (id_operatora)
139 /
140
141 CREATE INDEX IX_Relationship11 ON Klienci (id_adresu)
142 /
143
144 -- Add keys for table Klienci
145
146 ALTER TABLE Klienci ADD CONSTRAINT id_klienta_key PRIMARY KEY (id_klienta)
147 /
148
149 -- Table Telefony
150
151 CREATE TABLE Telefony(
152     id_urzadzenia Integer GENERATED ALWAYS AS IDENTITY(
153         START WITH 1
154         INCREMENT BY 1
155         NOMAXVALUE
156         NOMINVALUE
157         CACHE 20) NOT NULL,
158     typ_karty Varchar2(6 ) NOT NULL
159         CHECK ((typ_karty IN ('Micro', 'Mini', 'Nano'))),
160     pamiec Varchar2(30 ) NOT NULL,
161     eSIM Char(1 ) NOT NULL
162 )
163 /
164
165 -- Add keys for table Telefony
166

```

```

167 ALTER TABLE Telefony ADD CONSTRAINT Unique_Identifier6 PRIMARY KEY (id_urzadzenia)
168 /
169
170 -- Table Plany_komorkowe
171
172 CREATE TABLE Plany_komorkowe(
173     id_planu Integer GENERATED ALWAYS AS IDENTITY(
174         START WITH 1
175         INCREMENT BY 1
176         NOMAXVALUE
177         NOMINVALUE
178         CACHE 20) NOT NULL,
179     limit_internetu Integer NOT NULL,
180     limit_internetu_EU Integer NOT NULL,
181     limit_sms Integer NOT NULL,
182     limit_mms Integer NOT NULL,
183     roaming Integer
184 )
185 /
186
187 -- Add keys for table Plany_komorkowe
188
189 ALTER TABLE Plany_komorkowe ADD CONSTRAINT id_planu_key PRIMARY KEY (id_planu)
190 /
191
192 -- Table Oferty
193
194 CREATE TABLE Oferty(
195     id_oferty Integer GENERATED ALWAYS AS IDENTITY(
196         START WITH 1
197         NOMAXVALUE
198         NOMINVALUE
199         CACHE 20) NOT NULL,
200     nazwa Varchar2(100 ) NOT NULL,
201     okres Varchar2(30 ) NOT NULL,
202     cena Number(10,2) NOT NULL,
203     id_operatora Integer NOT NULL,
204     id_planu Integer NOT NULL
205 )
206 /
207
208 -- Create indexes for table Oferty
209
210 CREATE INDEX IX_oferuje_oferte ON Oferty (id_operatora)
211 /
212
213 CREATE INDEX IX_posiada_plan ON Oferty (id_planu)
214 /
215
216 -- Add keys for table Oferty
217
218 ALTER TABLE Oferty ADD CONSTRAINT id_oferty_key PRIMARY KEY (id_oferty)
219 /
220
221 -- Table Placowki
222

```

```

223 CREATE TABLE Placowki(
224     id_placowki Integer GENERATED ALWAYS AS IDENTITY(
225         START WITH 1
226         INCREMENT BY 1
227         NOMAXVALUE
228         NOMINVALUE
229         CACHE 20) NOT NULL,
230     telefon Varchar2(12 ) NOT NULL,
231     adres_email Varchar2(50 ) NOT NULL,
232     id_operatora Integer NOT NULL,
233     id_adresu Integer NOT NULL
234 )
235 /
236
237 -- Create indexes for table Placowki
238
239 CREATE INDEX IX_posiada_placowke ON Placowki (id_operatora)
240 /
241
242 CREATE INDEX IX_Relationship3 ON Placowki (id_adresu)
243 /
244
245 -- Add keys for table Placowki
246
247 ALTER TABLE Placowki ADD CONSTRAINT id_placowki_key PRIMARY KEY (id_placowki)
248 /
249
250 -- Table Serwisanci
251
252 CREATE TABLE Serwisanci(
253     id_pracownika Integer GENERATED ALWAYS AS IDENTITY(
254         START WITH 1
255         INCREMENT BY 1
256         NOMAXVALUE
257         NOMINVALUE
258         CACHE 20) NOT NULL,
259     prawo_jazdy Char(1 ) NOT NULL
260 )
261 /
262
263 -- Add keys for table Serwisanci
264
265 ALTER TABLE Serwisanci ADD CONSTRAINT Unique_Identifier14 PRIMARY KEY (id_pracownika)
266 /
267
268 -- Table Handlowce
269
270 CREATE TABLE Handlowce(
271     id_pracownika Integer GENERATED ALWAYS AS IDENTITY(
272         START WITH 1
273         INCREMENT BY 1
274         NOMAXVALUE
275         NOMINVALUE
276         CACHE 20) NOT NULL,
277     szkolenie_kasowe Char(1 ) NOT NULL
278 )

```



```

279 /
280
281 -- Add keys for table Handlowce
282
283 ALTER TABLE Handlowce ADD CONSTRAINT Unique_Identifier15 PRIMARY KEY (id_pracownika)
284 /
285
286 -- Table Urzadzenie
287
288 CREATE TABLE Urzadzenie(
289     id_urzadzenia Integer GENERATED ALWAYS AS IDENTITY(
290         START WITH 1
291         INCREMENT BY 1
292         NOMAXVALUE
293         NOMINVALUE
294         CACHE 20) NOT NULL,
295     nazwa Varchar2(50 ) NOT NULL,
296     model Varchar2(30 ) NOT NULL,
297     marka Varchar2(30 ) NOT NULL,
298     cena Number(10,2) NOT NULL
299 )
300 /
301
302 -- Add keys for table Urzadzenie
303
304 ALTER TABLE Urzadzenie ADD CONSTRAINT id_urzedzenia_key PRIMARY KEY (id_urzadzenia)
305 /
306
307 -- Table Routery
308
309 CREATE TABLE Routery(
310     id_urzadzenia Integer GENERATED ALWAYS AS IDENTITY(
311         START WITH 1
312         INCREMENT BY 1
313         NOMAXVALUE
314         NOMINVALUE
315         CACHE 20) NOT NULL,
316     liczba_anten Integer NOT NULL
317         CHECK ((liczba_anten IN (1, 2, 4, 8))),
318     predkosc Varchar2(15 ) NOT NULL,
319     czestotliwosc Varchar2(15 ) NOT NULL
320 )
321 /
322
323 -- Add keys for table Routery
324
325 ALTER TABLE Routery ADD CONSTRAINT Unique_Identifier17 PRIMARY KEY (id_urzadzenia)
326 /
327
328 -- Table Modemy
329
330 CREATE TABLE Modemy(
331     id_urzadzenia Integer GENERATED ALWAYS AS IDENTITY(
332         START WITH 1
333         INCREMENT BY 1
334         NOMAXVALUE

```

```

335     NOMINVALUE
336     CACHE 20) NOT NULL,
337     przeznaczenie Varchar2(3 ) NOT NULL
338         CHECK ((przeznaczenie IN ('3G','LTE', '4G', '5G'))),
339     intefejs Varchar2(7 ) NOT NULL
340         CHECK ((intefejs IN ('USB-C', 'USB-2.0'))),
341     czestotliwosc Varchar2(15 ) NOT NULL
342 )
343 /
344
345 -- Add keys for table Modemy
346
347 ALTER TABLE Modemy ADD CONSTRAINT Unique_Identifier18 PRIMARY KEY (id_urzadzenia)
348 /
349
350 -- Table Plany_komorkowe_Urzadzenie
351
352 CREATE TABLE Plany_komorkowe_Urzadzenie(
353     id_planu Integer GENERATED ALWAYS AS IDENTITY(
354         START WITH 1
355         INCREMENT BY 1
356         NOMAXVALUE
357         NOMINVALUE
358         CACHE 20) NOT NULL,
359     id_urzadzenia Integer NOT NULL
360 )
361 /
362
363 -- Table Wlasciciele
364
365 CREATE TABLE Wlasciciele(
366     id_wlasciciela Integer GENERATED ALWAYS AS IDENTITY(
367         START WITH 1
368         INCREMENT BY 1
369         NOMAXVALUE
370         NOMINVALUE
371         CACHE 20) NOT NULL,
372     imie Varchar2(20 ) NOT NULL,
373     nazwisko Varchar2(30 ) NOT NULL,
374     id_adresu Integer NOT NULL,
375     id_operatora Integer NOT NULL
376 )
377 /
378
379 -- Create indexes for table Wlasciciele
380
381 CREATE INDEX IX_wlasciciel_posiada_adres ON Wlasciciele (id_adresu)
382 /
383
384 CREATE INDEX IX_Relationship1 ON Wlasciciele (id_operatora)
385 /
386
387 -- Add keys for table Wlasciciele
388
389 ALTER TABLE Wlasciciele ADD CONSTRAINT PK_Wlasciciele PRIMARY KEY (id_wlasciciela)
390 /

```

```

391
392 -- Table Adresy
393
394 CREATE TABLE Adresy(
395     id_adresu Integer GENERATED ALWAYS AS IDENTITY(
396         START WITH 1
397         INCREMENT BY 1
398         NOMAXVALUE
399         NOMINVALUE
400         CACHE 20) NOT NULL,
401     Miasto Varchar2(20 ) NOT NULL,
402     ulica Varchar2(20 ) NOT NULL,
403     nr_domu Varchar2(5 ) NOT NULL,
404     id_poczty Integer NOT NULL
405 )
406 /
407
408 -- Create indexes for table Adresy
409
410 CREATE INDEX IX_Relationship2 ON Adresy (id_poczty)
411 /
412
413 -- Add keys for table Adresy
414
415 ALTER TABLE Adresy ADD CONSTRAINT PK_Adresy PRIMARY KEY (id_adresu)
416 /
417
418 -- Table Karty_SIM
419
420 CREATE TABLE Karty_SIM(
421     id_karty_SIM Integer GENERATED ALWAYS AS IDENTITY(
422         START WITH 1
423         INCREMENT BY 1
424         NOMAXVALUE
425         NOMINVALUE
426         CACHE 20) NOT NULL,
427     numer_telefonu Varchar2(12 ) NOT NULL,
428     rodzaj Varchar2(12 ) NOT NULL
429         CHECK ((rodzaj IN ('E-SIM','Phisical'))),
430     rozmiar Varchar2(15 ),
431     id_umowy Integer NOT NULL
432 )
433 /
434
435 -- Create indexes for table Karty_SIM
436
437 CREATE INDEX IX_dolaczenie_karty_SIM ON Karty_SIM (id_umowy)
438 /
439
440 -- Add keys for table Karty_SIM
441
442 ALTER TABLE Karty_SIM ADD CONSTRAINT PK_Karty_SIM PRIMARY KEY (id_karty_SIM)
443 /
444
445 -- Table Uprawnienia
446

```

```

447 CREATE TABLE Uprawnienia(
448     id_uprawnienia Integer GENERATED ALWAYS AS IDENTITY(
449         START WITH 1
450         INCREMENT BY 1
451         NOMAXVALUE
452         NOMINVALUE
453         CACHE 20) NOT NULL,
454     nazwa Char(20 ) NOT NULL,
455     id_pracownika Integer NOT NULL
456 )
457 /
458
459 -- Create indexes for table Uprawnienia
460
461 CREATE INDEX IX_ma_zawod ON Uprawnienia (id_pracownika)
462 /
463
464 -- Add keys for table Uprawnienia
465
466 ALTER TABLE Uprawnienia ADD CONSTRAINT PK_Uprawnienia PRIMARY KEY (id_uprawnienia)
467 /
468
469 -- Table Jezyki
470
471 CREATE TABLE Jezyki(
472     id_jezyku Integer GENERATED ALWAYS AS IDENTITY(
473         START WITH 1
474         INCREMENT BY 1
475         NOMAXVALUE
476         NOMINVALUE
477         CACHE 20) NOT NULL,
478     nazwa Varchar2(20 ) NOT NULL
479 )
480 /
481
482 -- Add keys for table Jezyki
483
484 ALTER TABLE Jezyki ADD CONSTRAINT PK_Jezyki PRIMARY KEY (id_jezyku)
485 /
486
487 ALTER TABLE Jezyki ADD CONSTRAINT nazwa UNIQUE (nazwa)
488 /
489
490 -- Table Poczty
491
492 CREATE TABLE Poczty(
493     id_poczty Integer GENERATED ALWAYS AS IDENTITY(
494         START WITH 1
495         INCREMENT BY 1
496         NOMAXVALUE
497         NOMINVALUE
498         CACHE 20) NOT NULL,
499     kod_poczty Char(6 ) NOT NULL,
500     poczta Varchar2(20 ) NOT NULL
501 )
502 /

```

```

503
504 -- Add keys for table Poczty
505
506 ALTER TABLE Poczty ADD CONSTRAINT PK_Poczty PRIMARY KEY (id_poczty)
507 /
508
509 ALTER TABLE Poczty ADD CONSTRAINT kod_poczty UNIQUE (kod_poczty)
510 /
511
512 -- Table and Columns comments section
513
514 COMMENT ON COLUMN Poczty.kod_poczty IS 'xx-xxx'
515 /
516
517 -- Table Stanowiska
518
519 CREATE TABLE Stanowiska(
520     id_stanowiska Integer GENERATED ALWAYS AS IDENTITY(
521         START WITH 1
522         INCREMENT BY 1
523         NOMAXVALUE
524         NOMINVALUE
525         CACHE 20) NOT NULL,
526     nazwa Varchar2(20 ) NOT NULL,
527     opis Varchar2(400 ) NOT NULL
528 )
529 /
530
531 -- Add keys for table Stanowiska
532
533 ALTER TABLE Stanowiska ADD CONSTRAINT PK_Stalowiska PRIMARY KEY (id_stanowiska)
534 /
535
536 ALTER TABLE Stanowiska ADD CONSTRAINT nazwa_1 UNIQUE (nazwa)
537 /
538
539 -- Table Znajomosc_jezykow
540
541 CREATE TABLE Znajomosc_jezykow(
542     id_pracownika Integer GENERATED ALWAYS AS IDENTITY(
543         START WITH 1
544         INCREMENT BY 1
545         NOMAXVALUE
546         NOMINVALUE
547         CACHE 20) NOT NULL,
548     id_jezyku Integer NOT NULL,
549     Kod_poziomu Char(2 ) NOT NULL
550         CHECK ((Kod_poziomu IN ('B1','B2','C1','C2','NS')))
551 )
552 /
553
554 -- Add keys for table Znajomosc_jezykow
555
556 ALTER TABLE Znajomosc_jezykow ADD CONSTRAINT PK_Znajomosc_jezykow PRIMARY KEY
557
558 (id_pracownika,id_jezyku)

```

```

559 /
560
561
562 -- Create foreign keys (relationships) section -----
563
564 ALTER TABLE Pracownicy ADD CONSTRAINT zatrudnia FOREIGN KEY (id_operatora)
565
566 REFERENCES Operatorzy (id_operatora)
567 /
568
569
570
571 ALTER TABLE Umowy ADD CONSTRAINT oferuje_umowe FOREIGN KEY (id_pracownika)
572
573 REFERENCES Pracownicy (id_pracownika)
574 /
575
576
577
578 ALTER TABLE Umowy ADD CONSTRAINT podpisuje_umowe FOREIGN KEY (id_klienta)
579
580 REFERENCES Klienci (id_klienta)
581 /
582
583
584
585 ALTER TABLE Pracownicy ADD CONSTRAINT nalezy_do_placowki FOREIGN KEY
586
587 (id_placowki) REFERENCES Placowki (id_placowki)
588 /
589
590
591
592 ALTER TABLE Oferty ADD CONSTRAINT oferuje_oferte FOREIGN KEY (id_operatora)
593
594 REFERENCES Operatorzy (id_operatora)
595 /
596
597
598
599 ALTER TABLE Placowki ADD CONSTRAINT posiada_placowke FOREIGN KEY (id_operatora)
600
601 REFERENCES Operatorzy (id_operatora)
602 /
603
604
605
606 ALTER TABLE Umowy ADD CONSTRAINT umowa_o_ofercie FOREIGN KEY (id_oferty)
607
608 REFERENCES Oferty (id_oferty)
609 /
610
611
612
613 ALTER TABLE Oferty ADD CONSTRAINT posiada_plan FOREIGN KEY (id_planu) REFERENCES
614

```

```

615 Plany_komorkowe (id_planu)
616 /
617
618
619
620 ALTER TABLE Klienci ADD CONSTRAINT posiada_klienta FOREIGN KEY (id_operatora)
621
622 REFERENCES Operatorzy (id_operatora)
623 /
624
625
626
627 ALTER TABLE Operatorzy ADD CONSTRAINT adres_biura_glownego FOREIGN KEY
628
629 (id_adresu) REFERENCES Adresy (id_adresu)
630 /
631
632
633
634 ALTER TABLE Wlasciciele ADD CONSTRAINT wlasciciel_posiada_adres FOREIGN KEY
635
636 (id_adresu) REFERENCES Adresy (id_adresu)
637 /
638
639
640
641 ALTER TABLE Pracownicy ADD CONSTRAINT pracownik_posiada_adres FOREIGN KEY
642
643 (id_adresu) REFERENCES Adresy (id_adresu)
644 /
645
646
647
648 ALTER TABLE Klienci ADD CONSTRAINT klient_posiada_adres FOREIGN KEY (id_adresu)
649
650 REFERENCES Adresy (id_adresu)
651 /
652
653
654
655 ALTER TABLE Uprawnienia ADD CONSTRAINT ma_zawod FOREIGN KEY (id_pracownika)
656
657 REFERENCES Serwisanci (id_pracownika)
658 /
659
660
661
662 ALTER TABLE Karty_SIM ADD CONSTRAINT dolaczenie_karty_SIM FOREIGN KEY (id_umowy)
663
664 REFERENCES Umowy (id_umowy)
665 /
666
667
668
669 ALTER TABLE Wlasciciele ADD CONSTRAINT Ma_wlasciciela FOREIGN KEY (id_operatora)
670

```

```

671 REFERENCES Operatorzy (id_operatora)
672 /
673
674
675
676 ALTER TABLE Adresy ADD CONSTRAINT Ma_poczte FOREIGN KEY (id_poczty) REFERENCES
677
678 Poczty (id_poczty)
679 /
680
681
682
683 ALTER TABLE Placowki ADD CONSTRAINT Placowka_ma_adres FOREIGN KEY (id_adresu)
684
685 REFERENCES Adresy (id_adresu)
686 /
687
688
689
690 ALTER TABLE Pracownicy ADD CONSTRAINT Ma_stanowisko FOREIGN KEY (id_stanowiska)
691
692 REFERENCES Stanowiska (id_stanowiska)
693 /
694
695
696
697 ALTER TABLE Znajomosc_jezykow ADD CONSTRAINT Zna_jezyk FOREIGN KEY
698
699 (id_pracownika) REFERENCES Handlowce (id_pracownika)
700 /
701
702
703
704 ALTER TABLE Znajomosc_jezykow ADD CONSTRAINT Jezyk_jest_znany FOREIGN KEY
705
706 (id_jezyku) REFERENCES Jezyki (id_jezyku)
707 /
708
709
710
711 ALTER TABLE Umowy ADD CONSTRAINT podpisuje_sie_w_placowce FOREIGN KEY
712
713 (id_placowki) REFERENCES Placowki (id_placowki)
714 /

```


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

5.4.1. Podgląd ofert tańszych niż 40 złotych

```
SELECT *  
FROM Oferty  
WHERE cena < 40;
```

| CENA | ID_OPERATORA | ID_PLANU |
|-------|--------------|----------|
| 29,99 | 37 | 26 |
| 39,99 | 38 | 27 |
| 19,99 | 40 | 29 |

5.4.2. Podgląd planów komórkowych z limitem internetu większym lub równym 20Gb

```
SELECT *  
FROM plany_komorkowe  
WHERE limit_internetu >= 20;
```

| ID_PLANU | LIMIT_INTERNETU | LIMIT_INTERNETU_EU | LIMIT_SMS | LIMIT_MMS | ROAMING |
|----------|-----------------|--------------------|-----------|-----------|---------|
| 29 | 20 | 15 | 400 | 200 | 0 |
| 30 | 25 | 20 | 500 | 250 | 1 |

5.4.3. Podgląd umów za październik 2023

```
SELECT *  
FROM Umowy  
WHERE EXTRACT(MONTH FROM data_podpisania) = 10  
AND EXTRACT(YEAR FROM data_podpisania) = 2023;
```

| ID_UMOWY | DATA_POD | ID_PRACOWNIKA | ID_KLIENTA | ID_OFERTY | ID_PLACOWKI |
|----------|----------|---------------|------------|-----------|-------------|
| 5 | 01.10.23 | 13 | 8 | 27 | 7 |
| 6 | 02.10.23 | 14 | 9 | 29 | 7 |

5.4.4. Podgląd klientów, którzy mieszkają w Białymstoku

```
SELECT Klienci.*
FROM Adresy
JOIN Klienci ON Adresy.id_adresu = Klienci.id_adresu
WHERE Adresy.miasto = 'Bialystok';
```

| ID_KLIENTA | IMIE | NAZWISKO |
|------------|-------|-----------|
| 7 | Anna | Jablonska |
| 8 | Anna | Wilczek |
| 11 | Jakub | Gac |

5.4.5. Podgląd serwisantów posiadających prawo jazdy

```
SELECT * FROM Serwisanci WHERE prawo_jazdy = 'T';
```

| ID_PRACOWNIKA | P |
|---------------|---|
| 1 | T |
| 4 | T |

5.4.6. Podgląd pracownika, który podpisał najwięcej umów w placówce o id_placówki = 7

```
SELECT Pracownicy.id_pracownika, Pracownicy.imie, Pracownicy.nazwisko, Pracownicy.id_placowki, COUNT(Umowy.id_umowy) AS liczba_umow
FROM Pracownicy
JOIN Umowy ON Pracownicy.id_pracownika = Umowy.id_pracownika
WHERE Pracownicy.id_placowki = 7
GROUP BY Pracownicy.id_pracownika, Pracownicy.imie, Pracownicy.nazwisko, Pracownicy.id_placowki
ORDER BY liczba_umow DESC
FETCH FIRST 1 ROW ONLY;
```

| ID_PRACOWNIKA | IMIE | NAZWISKO | ID_PLACOWKI | LICZBA_UMOW |
|---------------|--------|----------|-------------|-------------|
| 13 | Maciej | Marowski | 7 | 2 |

6. Bibliografia

1. Materiały wykładowe dr hab. inż. Marcina Kowalczyka, Politechnika Warszawska, wydział Elektroniki i Technik Informacyjnych, kierunek Telekomunikacja, przedmiot Bazy danych i Big Data.
2. Projektowanie i normalizacja bazy danych - Jakub Kasprzak.
<https://www.sqlpedia.pl/projektowanie-i-normalizacja-bazy-danych/>