

xxx
xxx

Dostawca energii miejskiej

Spis treści projektu:

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

2. Definicja systemu

2.1 Perspektywy użytkowników

3. Model konceptualny

3.1 Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

3.2 Ustalenie związków między encjami i ich typów

3.3 Określenie atrybutów i ich dziedzin

3.4 Dodatkowe reguły integralnościowe (reguły biznesowe)

3.5 Klucze kandydujące i główne (decyzje projektowe)

3.6 Schemat ER na poziomie konceptualnym

3.7 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

4. Model logiczny

4.1 Charakterystyka modelu relacyjnego

4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

4.3 Proces normalizacji – analiza i przykłady

4.4 Schemat ER na poziomie modelu logicznego

4.5 Więzy integralności

4.6 Proces denormalizacji – analiza i przykłady

5. Faza fizyczna

5.1 Projekt transakcji i weryfikacja ich wykonalności

5.2 Strojenie bazy danych – dobór indeksów

5.3 Skrypt SQL zakładający bazę danych

5.4 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

Bibliografia

1. Zakres i cel projektu - Opis działalności biznesowej

Celem tego projektu jest zaprojektowanie, a następnie implementacja relacyjnej bazy danych ściśle związanej z dostawcami energii miejskiej. Realizacja tego projektu podzielona została na trzy fazy; konceptualną, logiczną oraz fizyczną. Każda z nich zostanie szczegółowo przedstawiona i wyjaśniona w poniższej dokumentacji.

Dostawcy energii miejskiej - Realizowany projekt dotyczy stworzenia bazy danych dla przedsiębiorstwa jakim jest dostawca energii miejskiej. Przedsiębiorstwo to ma świadczyć usługi

2. Definicja systemu

2.1 Perspektywy użytkowników

Baza danych dostawcy energii miejskiej będzie wykorzystywana przez następujących użytkowników systemu:

Właściciele - osoba, której firma jest własnością. Ma dostęp do wszystkich danych z nią związanych.

Kierownicy - osoba, pod której nadzorem znajduje się dany oddział firmy.

Pracownicy - pracownik firmy. Ma bezpośredni kontakt z klientem. Ma wgląd do swoich danych jako pracownika, do danych obsługiwanych klientów oraz do dodawania nowych transakcji..

Klienci - osoba, która kupuje usługę, energię elektryczną od dostawcy. Ma bezpośredni kontakt z oddziałem firmy.

2.2 Funkcjonalności bazy danych:

Uprawnienia	Właściciel	Kierownik	Pracownik	Klient
Podgląd do informacji o dostawcy	tak	tak	tak	tak
Modyfikowanie informacji o dostawcy	tak	nie	nie	nie
Usuwanie informacji o dostawcy	tak	nie	nie	nie
Podgląd informacji o pracownikach	tak	tak	nie	nie
Podgląd informacji do danych kontaktowych pracownika	tak	tak	tak	tak
Modyfikowanie informacji o pracownikach	tak	tak	nie	nie
Usuwanie informacji o pracownikach	tak	tak	nie	nie
Podgląd do informacji o usługach	tak	tak	tak	tak
Modyfikowanie informacji o usługach	tak	nie	nie	nie
Usuwanie informacji o usługach	tak	nie	nie	nie
Podgląd do informacji o kliencie	tak	tak	tak	nie
Modyfikowanie informacji o kliencie	tak	tak	tak	nie
Usuwanie informacji o kliencie	tak	tak	tak	nie

Podgląd do informacji o infrastrukturze dostawcy	tak	tak	tak	nie
Modyfikowanie informacji o infrastrukturze dostawcy	tak	nie	nie	nie
Usuwanie informacji o infrastrukturze dostawcy	tak	nie	nie	nie

3. Model konceptualny

3.1 Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

Na poziomie konceptualnym wyróżniliśmy następujące encje:

- **Dostawca(Firma)** - główna encja firmy, dostawcy energii miejskiej, zawiera szczegółowe informacje na jej temat niezbędne do prowadzenia działalności, takie jak (nazwa, adres, numer telefonu).
- **Oddział** - jednostka reprezentatywna firmy, obsługująca klientów i zatrudniająca pracowników
- **Usługa** - oferowana przez dostawcę dla klientów, realizowana przez pracowników firmy
- **Pracownik** - encja opisująca pracowników firmy.
 - **Inżynier** - osoba z konkretną specjalizacją, jest bezpośrednim wykonawcą usług montażowych jak i naprawczych
 - **Konsultant** - osoba posiadająca stały kontakt z klientem
- **Klient** - encja opisująca klientów firmy, osoby które kupują usługę dostawcy energii miejskiej
 - **Firma** - podmiot w ramach, którego przedsiębiorca prowadzi działalność gospodarczą lub zawodową

- **Osoba prywatna** - osoba kupująca usługę dla gospodarstwa domowego
- **Infrastruktura** - encja zawierająca wszystkie istotne elementy fizyczne zaopatrzenia dostawcy energii miejskiej

3.2 Ustalenie związków między encjami i ich typów

Nazwa relacji	Krotność	Typ uczestnictwa	Opis
Dostawca - Oddział	1..1 - 0..M	Dostawca nie musi otwierać żadnego oddziału, jednak w celu rozwoju firmy otwiera wiele oddziałów. Natomiast oddziały otwierane są przez jednego i tylko jednego dostawcę.	Dostawca otwiera oddział, natomiast oddział jest otwierany przez dostawcę
Dostawca - Infrastruktura	1..1 - 1..M	Dostawca aby jego firma mogła funkcjonować musi posiadać jakąś infrastrukturę, którą samemu wybudowuje. Infrastruktura należy natomiast do jednego i tylko jednego dostawcy.	Dostawca buduje infrastrukturę, a infrastruktura jest budowana przez dostawcę.
Dostawca - Usługa	1..1 - 1..M	Dostawca musi oferować jakiejkolwiek usługi, aby jego firma mogła funkcjonować. Usługa natomiast jest oferowana przez jednego i tylko jednego dostawcę.	Dostawca oferuje usługę, a usługa jest oferowana przez dostawcę.
Oddział - Pracownik	1..N - 0..M	Oddział może, jednak nie musi zatrudniać pracowników. W celu rozwoju firmy jest to bardzo ważne, szczególnie w kontekście wykonywania usług. Natomiast pracownicy mogą być zatrudniani przez wiele oddziałów.	Oddział zatrudnia pracownika, a pracownik jest zatrudniany przez oddział

Oddział - Infrastruktura	1..N - 0..M	Oddział może, jednak nie musi znajdować się w miejscu gdzie istnieje infrastruktura dostawcy, którą w takim wypadku dany oddział miałby się zajmować. Infrastruktura natomiast musi być serwisowana przez minimum jeden oddział, a może nawet przez wiele.	Oddział serwisuje infrastrukturę, a infrastruktura jest serwisowana przez oddział.
Oddział - Klient	1..N - 0..M	Oddział może, ale nie musi posiadać klienta których również może być wielu. Klient natomiast musi być obsługiwany przez któryś z oddziałów.	Oddział obsługuje klienta, natomiast klient jest obsługiwany przez oddział.
Oddział - Usługa	1..N - 0..M	Oddział może nie otrzymywać zleceń od klientów, zarówno jak i może otrzymywać ich wiele. Usługa za to musi być przyjmowana przez jakiś z oddziałów.	Oddział przyjmuje zlecenia na wykonanie usługi, natomiast usługi są przyjmowane przez oddziały.
Pracownik - Usługa	1..N - 0..M	Pracownik może wykonywać usługi, jednak możliwy jest również scenariusz, w którym dany pracownik(których może być wielu) nie wykonuje danej usługi(może to wynikać np. z specjalizacji).	Pracownik wykonuje usługę, a usługa jest wykonywana przez pracownika.
Pracownik- Inżynier	-	-	Inżynier jako specjalizacja encji Pracownik

Pracownik-Konsultant	-	-	Konsultant jako specjalizacja encji Pracownik
Klient - Osoba prywatna	-	-	Osoba prywatna jako specjalizacja encji Klient
Klient - Firma	-	-	Firma jako specjalizacja encji Klient

3.3 Określenie atrybutów i ich dziedzin

Dostawca			
ATRYBUT	TYP I DZIEDZINA	OBOWIĄZKO- WOŚĆ	OPIS
ID_Dostawcy	SmallInt	Obowiązkowy	Unikatowy identyfikator dostawcy/firmy
Nazwa_Dostawcy	VarChar(200)	Obowiązkowy	Nazwa Dostawcy Energii Miejskiej
Adres	VarChar(400)	Obowiązkowy	Adres głównej siedziby firmy
Data_zalozenia	Date	Obowiązkowy	Data założenia firmy
Wlasciciel	VarChar(600)	Obowiązkowy	Właściciel firmy
NIP	VarChar(13)	Obowiązkowy	Numer identyfikacji podatkowej firmy

KRS	VarChar(10)	Obowiązkowy	Krajowy Rejestr Sądowy firmy
REGON	VarChar(14)	Obowiązkowy	Identyfikator REGON firmy
Strona_internetowa	VarChar(40)	Obowiązkowy	Strona internetowa firmy
Nr_telefonu	VarChar(15)	Obowiązkowy	Numer telefonu firmy
Email	VarChar(30)	Obowiązkowy	Adres email firmy

Oddział			
ATRYBUT	TYP I DZIEDZINA	OBOWIĄZKO- WOŚĆ	OPIS
ID_oddialu	SmallInt	Obowiązkowy	Unikatowy identyfikator biura
Adres	VarChar(400)	Obowiązkowy	Adres oddziału, pole segmentowe (ulica, nr lokalu, miasto, kod pocztowy, poczta)
Data_zalozenia	Date	Obowiązkowy	Data założenia oddziału
Kierownik	VarChar(600)	Obowiązkowy	Kierownik oddziału

PRACOWNIK			
ATRYBUT	TYP I DZIEDZINA	OBOWIĄZKO- WOŚĆ	OPIS
ID_pracownika	Integer	Obowiązkowy	Unikatowy identyfikator pracownika
Imie	VarChar(20)	Obowiązkowy	Imię pracownika

Nazwisko	VarChar(20)	Obowiązkowy	Nazwisko pracownika
Data_urodzenia	Date	Obowiązkowy	Data urodzenia pracownika
PESEL	Character(11)	Nie**	Numer PESEL pracownika
Plec	Character(1), PlecR('K', 'M')*	Obowiązkowy	Płeć pracownika
Adres	VarChar(400)	Obowiązkowy	Adres pracownika, pole segmentowe (ulica, id_oddzialu, miasto, kod pocztowy, poczta)
Stanowisko	VarChar(20)	Obowiązkowy	Stanowisko, na którym jest zatrudniony pracownik
Data_zatrudnienia	Date	Obowiązkowy	Data zatrudnienia pracownika
Nr_konta	Character(26)	Obowiązkowy	Numer konta bankowego w standardzie IBAN, na które przelana jest wpłata
Email	VarChar(30)	Obowiązkowy	Adres e-mail pracownika
Nr_telefonu	VarChar(15)	Obowiązkowy	Służbowy numer telefonu pracownika
Wykształcenie	VarChar(400)	Obowiązkowy	Wykształcenie pracownika
Wynagrodzenie	Number	Obowiązkowy	Stawka miesięcznego wynagrodzenia dla pracownika

Uzasadnienie:

*Płeć to pole, które opisuje cechę należącą do skończonego zbioru możliwości z dziedziny K, M. Wykorzystuje odpowiadającą jej regułę.

**Oddział firmy jest otwarty na zatrudnianie pracowników nie mających polskiego obywatelstwa.

Inżynier			
ATRYBUT	TYP I DZIEDZINA	OBOWIĄZKO- WOŚĆ	OPIS
Prawo_jazdy	Boolean	Obowiązkowy	Czy inżynier posiada prawo jazdy
Specjalizacja	VarChar(400)	Obowiązkowy	Nazwa specjalizacji

Konsultant			
ATRYBUT	TYP I DZIEDZINA	OBOWIĄZKO- WOŚĆ	OPIS
Ilosc_znanych_jezykow	SmallInt	Obowiązkowy	Ilość znanych przez pracownika języków na poziomie przynajmniej B2
Mozliwosc_pracy_zdalnej	Boolean	Obowiązkowy	Informacja o możliwości pracy konsultanta z miejsca zamieszkania

KLIENT			
ATRYBUT	TYP I DZIEDZINA	OBOWIĄZKO- WOŚĆ	OPIS
ID_klienta	Integer	Obowiązkowy	Unikatowy identyfikator klienta
Imie	VarChar(20)	Obowiązkowy	Imię klienta
Nazwisko	VarChar(30)	Obowiązkowy	Nazwisko klienta
Adres	VarChar(400)	Obowiązkowy	Adres klienta, pole segmentowe (ulica, nr lokalu, miasto, kod pocztowy, poczta)
Nr_telefonu	VarChar(15)	Obowiązkowy	Numer telefonu klienta
Email	VarChar(30)	Obowiązkowy	Adres e-mail klienta
Zuzycie_pradu(kWh)	Number	Obowiązkowy	Ilość średnio zużytego prądu w skali roku
Data_zawarcia_umowy	Date	Obowiązkowy	Data zawarcia umowy z dostawcą
Data_zakończenia_umowy	Date	Nie*	Data zakończenia umowy z dostawcą
PESEL	Character(11)	Nie**	Numer PESEL klienta
Nr_konta	Character(26)	Obowiązkowy	Numer konta bankowego klienta w standardzie IBAN

Uzasadnienie:

*Płeć to pole, które opisuje cechę należącą do skończonego zbioru możliwości z dziedziny K,M Wykorzystuje odpowiadającą jej regułę.

**Klientami firmy mogą być osoby nieposiadające polskiego obywatelstwa

Firma			
ATRYBUT	TYP I DZIEDZINA	OBOWIĄZKO- WOŚĆ	OPIS
Nazwa_przedsiębiorstwa	VarChar(400)	Obowiązkowy	Nazwa firmy
NIP	VarChar(13)	Obowiązkowy	Numer identyfikacji podatkowej firmy
KRS	VarChar(10)	Obowiązkowy	Krajowy Rejestr Sądowy firmy
REGON	VarChar(14)	Obowiązkowy	Identyfikator REGON firmy
Przedstawiciel	VarChar(40)	Obowiązkowy	Imię i nazwisko przedstawiciela firmy

Osoba_prywatna			
ATRYBUT	TYP I DZIEDZINA	OBOWIĄZKO- WOŚĆ	OPIS
Typ_budynku	VarChar(40)	Obowiązkowy	Typ budynku
Karta_duzej_rodziny	Boolean	Obowiązkowy	Czy posiada kartę dużej rodziny

Usługa			
ATRYBUT	TYP I DZIEDZINA	OBOWIĄZKO- WOŚĆ	OPIS
ID_uslugi	Integer	Obowiązkowy	Unikatowy identyfikator usługi

Rodzaj_usługi	VarChar(200)	Obowiązkowy	Rodzaj świadczonej usługi
Cena	Number	Obowiązkowy	Wartość usługi w złotych
Data realizacji	Date	Obowiązkowy	Data realizacji usługi
Pochodzenie_pradu	VarChar(30)	Nie*	Z jakiej elektrowni pochodzi prąd

Uzasadnienie:

*Pochodzenie_pradu nie jest to informacja obowiązkowa do podania przez dostawcę, gdyż pochodzenie prądu może być podzielone na udział procentowy danych elektrowni, jak również większości klientów nie interesują tak dokładne dane.

Infrastruktura			
ATRYBUT	TYP I DZIEDZINA	OBOWIĄZKOWOŚĆ	OPIS
ID_elementu_infrastruktur_y	SmallInt	Obowiązkowy	Unikatowy identyfikator nieruchomości
Polozenie	VarChar(400)	Obowiązkowy	Położenie
Typ_elementu_elektronicznego	VarChar(40)	Obowiązkowy	Typ elementu, model
Data_produkcji	Date	Obowiązkowy	Data wyprodukowania elementu
Data_ostatniego_serwisu	Date	Obowiązkowy	Data ostatnio przeprowadzanego serwisu
Producent	VarChar(40)	Obowiązkowy	Nazwa producenta elementu
Gwarancja	Boolean	Obowiązkowy	Czy posiada gwarancję

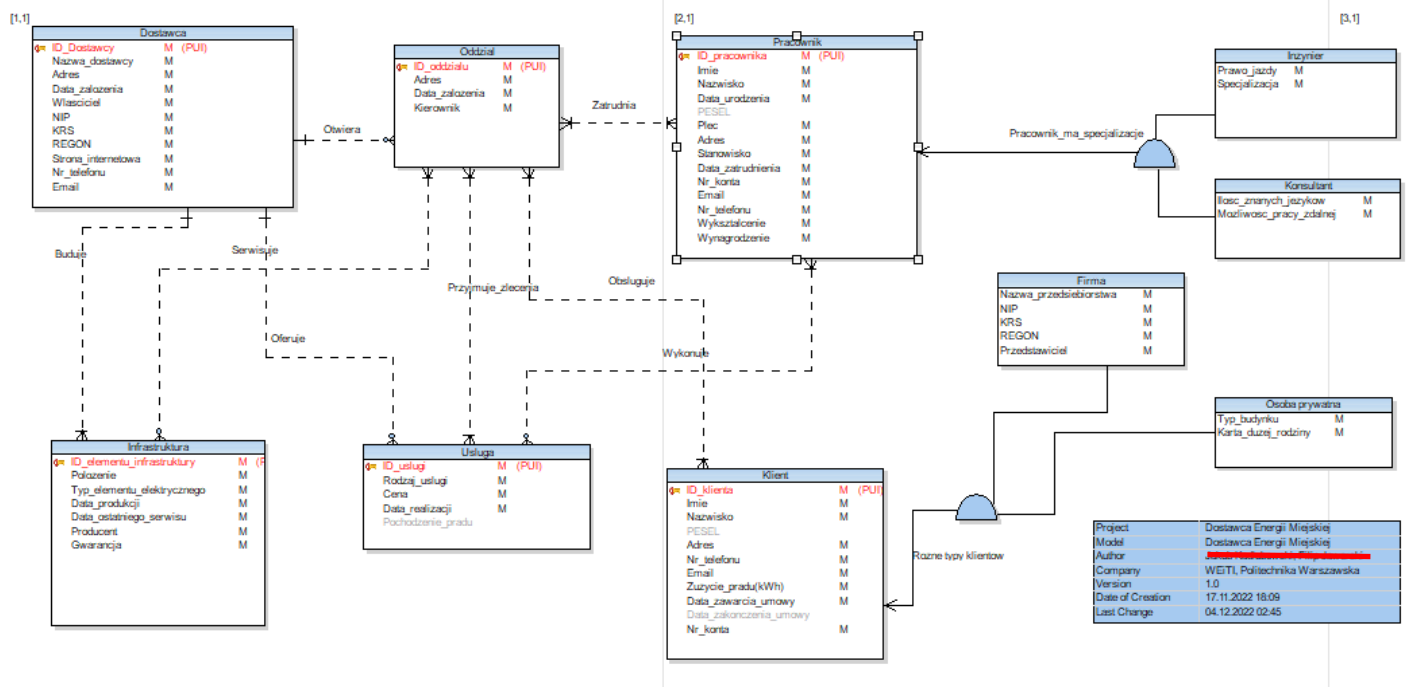
3.4 Dodatkowe reguły integralnościowe (reguły biznesowe)

- Dana usługa może być wielokrotnie sprzedawana wielu różnym klientom. Nie oznacza to, że otrzymują oni jeden produkt do podziału, a odrębne i samodzielne zasoby w postaci energii elektrycznej do samodzielnego rozdysponowania.
- Encja oddział stworzona została jako oddzielna encja, gdyż nie stanowi rdzenia przedsiębiorstwa i nie dziedziczy atrybutów dostawcy. Stanowi bezpośrednie połączenie z klientem będąc równocześnie reprezentatywnym elementem przedsiębiorstwa.
- Zakładamy, że klient zainteresowany usługą dostawcy energii miejskiej skontaktuje się z oddziałem firmy, a nie bezpośrednio z samym dostawcą

3.5 Klucze kandydujące i główne (decyzje projektowe)

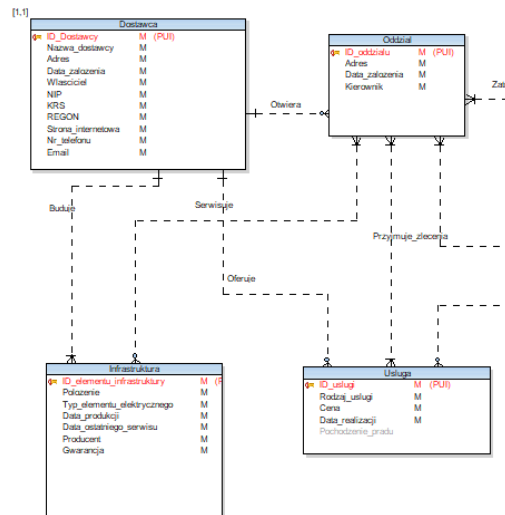
Nazwa encji	Klucz główny	Klucz kandydujący
Dostawca	ID_Dostawcy	Nazwa_dostawcy
Oddział	ID_oddzialu	Kierownik
Pracownik	ID_pracownika	PESEL, Nr_konta
Inżynier	ID_pracownika	PESEL, Nr_konta
Konsultant	ID_pracownika	PESEL, Nr_konta
Usługa	ID_usługi	Typ_elementu_elektronicznego
Klient	ID_klienta	PESEL, Nr_konta, Nr_telefonu, Email
Osoba_prywatna	ID_klienta	PESEL, Nr_konta, Nr_telefonu, Email
Firma	ID_klienta	PESEL, Nr_konta, Nr_telefonu, Email

3.6 Schemat ER na poziomie konceptualnym



3.7 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

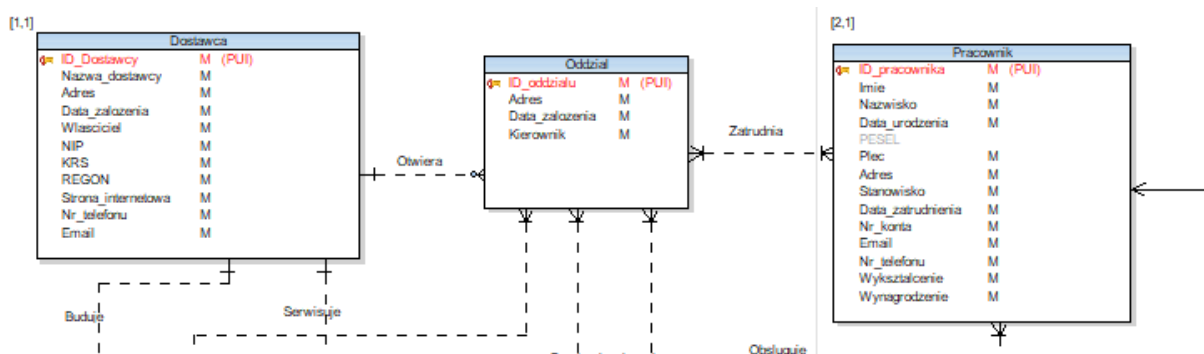
1. Wystąpienie pułapki wachlarzowej



Możliwe wystąpienie pułapki wachlarzowej w naszym modelu

Tworząc encję 'Infrastruktura' pomyśleliśmy o dwóch różnych relacjach jakie chcielibyśmy do niej prowadzić. By uniknąć pułapki wachlarzowej, w której encja 'Dostawca' posiadałaby dwie relacje: 'Buduje' i 'Serwisuje', stworzyliśmy oddzielną encję 'Oddział', która odciąża ją z relacji 'Serwisuje' jak i z relacji 'Przyjmuje zlecenia'.

2. Pułapka szczelinowa



Analizując nasz model projektowy zauważyliśmy miejsce występowanie potencjalnej pułapki szczelinowej.

Występowałaby ona między encjami 'Dostawca', a 'Pracownik' połączonymi z encją 'Oddział', gdyby istniała możliwość zatrudnienia

pracownika bezpośrednio do dostawcy, a nie do oddziału. Nasz model nie zakłada jednak takiej sytuacji i jest to sprzeczne z przyjętym modelem biznesowym, więc pułapka szczelinowa nie występuje.

4. Model logiczny

4.1 Charakterystyka modelu relacyjnego

Po zakończeniu fazy koncektualnej projektowania naszej bazy danych dokonaliśmy konwersji projektu do poziomu logicznego. W skutek tej operacji oprogramowanie *Toad Data Modeler* dokonało następujących zmiany:

- Relacje “wielu do wielu” zastąpione zostały dwiema relacjami “jeden do wielu” które łączy tabela. Oczywiście tym samym, nowe relacje po obydwu stronach mają obowiązkowy charakter uczestnictwa.
- Typy danych zostały przekształcone do tych, które używa wybrany przez nas silnik bazy danych (Oracle 19c).
- Stworzone przez nas klucze główne każdej z encji zostały zaznaczone kolorem czerwonym.
- Encje otrzymały nowe atrybuty będące kluczami obcymi zaznaczonymi kolorem zielonym.
- Specjalizacje zyskały nowy atrybut w postaci klucza głównego swojego rodzica zaznaczonego kolorem granatowym.
- Powstały nowe relacje między tabelami uszczegóławiający, a ich encjami nadrzędnymi. Typ tych relacji jest określony jako “jeden do wielu” z nieobowiązkowym typem uczestnictwa po stronie potomka.
- Zmianie uległ kształt tabeli łączących oraz tabeli specjalizacji na prostokąty o bardziej zaokrąglonych wierzchołkach.

4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

Wykonaliśmy następujące operacje:

- **usunięcie związków wiele do wielu** - tabele 'Oddzial_Infrastruktura', 'Oddzial_Usluga', 'Pracownik_Usluga', 'Oddzial_Klient', 'Oddzial_Pracownik'.
- **zmiana nazw encji z liczby pojedynczej na mnogą** - 'Dostawca' -> 'Dostawcy', 'Oddzial' -> 'Oddzialy', 'Adres' -> 'Adresy' itd.
- **usunięcie pól wielowartościowych** - tabele 'Wlasciciele', 'Adresy', 'Kierownicy', 'Stanowiska', 'Wynagrodzenia', 'Poczty'.

4.3 Proces normalizacji – analiza i przykłady

Normalizacja to proces organizowania danych w bazie danych. Obejmuje to tworzenie tabel i ustanawianie relacji między tymi tabelami zgodnie z regułami opracowanymi w celu zarówno ochrony danych, jak i zapewnienia większej elastyczności bazy danych przez wyeliminowanie nadmiarowości i niespójnych zależności. Podczas normalizacji bazy danych powinniśmy wziąć pod uwagę kryteria, które definiują jak dobrze baza danych odwzorowuje tę przystępną formę.

Pierwsza postać normalna (**1NF**) jest osiągnięta, gdy każda wartość w bazie danych jest polem atomowym oraz gdy jest brak powtarzających się grup, takich jak np. 'Adres', 'poczta', czy 'stanowisko'). By to osiągnąć wykonaliśmy następujące działania:

- usunęliśmy pola 'Adres' występujące w relacjach 'Dostawcy', 'Oddzialy', 'Klienci', 'Pracownicy', 'Infrastruktury', następnie stworzyliśmy tablicę 'Adres' i połączyliśmy ją ze wspomnianymi encjami
- usunęliśmy pole wielowartościowe 'Wlasciciel' z relacji 'Dostawcy' i stworzyliśmy w zamian za to relację słownikową 'Wlasciciele'.
- To samo zrobiliśmy z polami: 'Kierownik' w relacji 'Oddzialy' i 'Wynagrodzenia', 'Stanowiska' w relacji 'Pracownicy', oraz 'Poczty' w relacji 'Adresy'.

Druga postać normalna (**2NF**) jest osiągana, gdy spełniony jest pierwszy postulat Codd'a oraz, gdy wszystkie wartości spoza klucza zależą w całości od klucza. Dzięki zastosowaniu w encjach kluczy prostych poprzez stworzenie kluczy sztucznych ('ID'), automatycznie osiągnęliśmy 2NF.

Trzecia postać normalna (**3NF**) wymaga osiągnięcia poprzednich dwóch oraz zapewnienia sytuacji, w której każdy atrybut danej relacji nie wchodzący w skład żadnego klucza potencjalnego nie był przechodnio funkcyjnie zależny od żadnego klucza potencjalnego tej relacji, a więc był określony tylko i wyłącznie kluczem. Ten stan spójności również udało się osiągnąć automatycznie, ponieważ wszystkie niekluczowe atrybuty w krotce mogą być zidentyfikowane dzięki prostemu kluczowi głównemu, a cały model został możliwie jak najbardziej uproszczony i rozłożony na integralne elementy trafnie i precyzyjnie reprezentujące pewien określony obiekt.



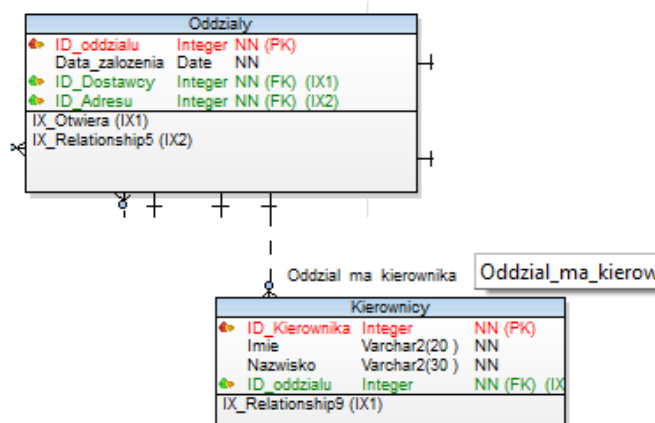
4.5 Więzy integralności

Aby spełnić warunki integralności danych, należy zdefiniować więzy integralności. W naszym projekcie praktycznie każdy atrybut jest polem obowiązkowym, tylko nieznaczna ilość atrybutów może przyjąć wartość NULL, np. 'Data_zakonczenia_umowy' czy 'PESEL'.

4.6 Proces denormalizacji – analiza i przykłady

Proces denormalizacji jest procesem będącym odwrotnością normalizacji wykonywanym w celu osiągnięcia szybszego zwracania odpowiedzi oraz ułatwienia dostępu do poszczególnych danych. Takie działanie wiąże się z kosztem w postaci zwiększenia rozmiaru bazy danych i większego ryzyka utraty ich spójności.

Postanowiliśmy nie korzystać z denormalizacji w naszej bazie danych. Stwierdziliśmy, że nie jest to konieczne, ponieważ dostęp do wszystkich danych jest łatwy. Gdybyśmy jednak chcieli się na to zdecydować, potencjalnym miejscem w którym dokonalibyśmy denormalizacji byłyby poniższe tabele, gdyż z relacji 'Kierownicy' korzysta tylko relacja 'Oddzialy'.



5. Faza fizyczna

5.1 Projekt transakcji i weryfikacja ich wykonalności

Uprawnienia	Właściciel	Kierownik	Pracownik	Klient	Czy wykonywalna?
Podgląd do informacji o dostawcy	tak	tak	tak	tak	tak
Modyfikowanie informacji o dostawcy	tak	nie	nie	nie	tak
Usuwanie informacji o dostawcy	tak	nie	nie	nie	tak
Podgląd informacji o pracownikach	tak	tak	nie	nie	tak
Podgląd informacji do danych kontaktowych pracownika	tak	tak	tak	tak	tak
Modyfikowanie informacji o pracownikach	tak	tak	nie	nie	tak
Usuwanie informacji o pracownikach	tak	tak	nie	nie	tak
Podgląd do informacji o usługach	tak	tak	tak	tak	tak
Modyfikowanie informacji o usługach	tak	nie	nie	nie	tak

Usuwanie informacji o usługach	tak	nie	nie	nie	tak
Podgląd do informacji o kliencie	tak	tak	tak	nie	tak
Modyfikowanie informacji o kliencie	tak	tak	tak	nie	tak
Usuwanie informacji o kliencie	tak	tak	tak	nie	tak
Podgląd do informacji o infrastrukturze dostawcy	tak	tak	tak	nie	tak
Modyfikowanie informacji o infrastrukturze dostawcy	tak	nie	nie	nie	tak
Usuwanie informacji o infrastrukturze dostawcy	tak	nie	nie	nie	tak

5.2 Strojenie bazy danych – dobór indeksów

Do strojenia bazy danych posłużą nam indeksy wygenerowane automatycznie w skrypcie SQL zakładającym bazę:

```
CREATE INDEX IX_Relationship2 ON Dostawcy (ID_Adresu)
CREATE INDEX IX_Otwiera ON Oddzialy (ID_Dostawcy)
CREATE INDEX IX_Relationship5 ON Oddzialy (ID_Adresu)
CREATE INDEX IX_Relationship3 ON Pracownicy (ID_Adresu)
CREATE INDEX IX_Relationship1 ON Pracownicy (ID_Stanowiska)
CREATE INDEX IX_Relationship4 ON Klienci (ID_Adresu)
CREATE INDEX IX_Oferuje ON Uslugi (ID_Dostawcy)
CREATE INDEX IX_Buduje ON Infrastruktury (ID_Dostawcy)
CREATE INDEX IX_Relationship6 ON Infrastruktury (ID_Adresu)
```



```

CREATE INDEX IX_Relationship10 ON Adresy (ID_Poczty)
CREATE INDEX IX_Relationship8 ON Wlasciciele (ID_Dostawcy)
CREATE INDEX IX_Relationship9 ON Kierownicy (ID_oddzialu)
CREATE INDEX IX_Relationship31 ON Wynagrodzenia (ID_pracownika)

```

5.3 Skrypt SQL zakładający bazę danych

```

/*
Created: 03.12.2022
Modified: 05.12.2022
Project: Dostawca Energii Miejskiej
Model: Dostawca Energii Miejskiej
Company: WEiT, Politechnika Warszawska
Author: Jakub Kadłubowski, Filip Jaworski
Version: 1.0
Database: Oracle 19c
*/

-- Create sequences section
-----

CREATE SEQUENCE DostawcaSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE OddzialSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE PracownikSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE StanowiskoSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE

```

```

    CACHE 20
/

CREATE SEQUENCE WynagrodzenieSeq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE KlientSeq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE WlascicielSeq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE InfrastrukturaSeq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE AdresSeq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE UsługaSeq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

```

```

CREATE SEQUENCE KierownikSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE PocztaSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

-- Create tables section -----

-- Table Dostawcy

CREATE TABLE Dostawcy(
  ID_Dostawcy Integer NOT NULL,
  Nazwa_dostawcy Varchar2(200 ) NOT NULL,
  Data_zalozenia Date NOT NULL,
  NIP Varchar2(13 ) NOT NULL,
  KRS Varchar2(10 ) NOT NULL,
  REGON Varchar2(14 ) NOT NULL,
  Strona_internetowa Varchar2(40 ) NOT NULL,
  Nr_telefonu Varchar2(15 ) NOT NULL,
  Email Varchar2(40 ) NOT NULL,
  ID_Adresu Integer NOT NULL
)
/

-- Create indexes for table Dostawcy

CREATE INDEX IX_Relationship2 ON Dostawcy (ID_Adresu)
/

-- Add keys for table Dostawcy

ALTER TABLE Dostawcy ADD CONSTRAINT Unique_Identifier1 PRIMARY KEY
(ID_Dostawcy)
/

-- Table Oddzialy

CREATE TABLE Oddzialy(
  ID_oddzialu Integer NOT NULL,
  Data_zalozenia Date NOT NULL,
  ID_Dostawcy Integer NOT NULL,

```

```

        ID_Adresu Integer NOT NULL
    )
    /

-- Create indexes for table Oddzialy

CREATE INDEX IX_Otwiera ON Oddzialy (ID_Dostawcy)
/

CREATE INDEX IX_Relationship5 ON Oddzialy (ID_Adresu)
/

-- Add keys for table Oddzialy

ALTER TABLE Oddzialy ADD CONSTRAINT Unique_Identifier2 PRIMARY KEY
(ID_oddzialu)
/

-- Table Pracownicy

CREATE TABLE Pracownicy(
    ID_pracownika Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(20 ) NOT NULL,
    Data_urodzenia Date NOT NULL,
    PESEL Char(11 ),
    Plec Char(256 ) NOT NULL,
    Data_zatrudnienia Date NOT NULL,
    Nr_konta Varchar2(20 ) NOT NULL,
    Email Varchar2(40 ) NOT NULL,
    Nr_telefonu Varchar2(15 ) NOT NULL,
    Wyksztalcenie Varchar2(400 ) NOT NULL,
    ID_Adresu Integer NOT NULL,
    ID_Stnowiska Integer NOT NULL
)
/

-- Create indexes for table Pracownicy

CREATE INDEX IX_Relationship3 ON Pracownicy (ID_Adresu)
/

CREATE INDEX IX_Relationship1 ON Pracownicy (ID_Stnowiska)
/

-- Add keys for table Pracownicy

ALTER TABLE Pracownicy ADD CONSTRAINT Unique_Identifier3 PRIMARY KEY
(ID_pracownika)
/

```

```

-- Table Klienci

CREATE TABLE Klienci(
    ID_klienta Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(20 ) NOT NULL,
    PESEL Char(11 BYTE) NOT NULL,
    Nr_telefonu Varchar2(15 ) NOT NULL,
    Email Varchar2(40 ),
    Nr_konta Varchar2(26 ) NOT NULL,
    Zuzycie_pradu(kWh) Number(8,2) NOT NULL,
    Data_zawarcia_umowy Date NOT NULL,
    Data_zakonczenia_umowy Date,
    ID_Adresu Integer NOT NULL
)
/

-- Create indexes for table Klienci

CREATE INDEX IX_Relationship4 ON Klienci (ID_Adresu)
/

-- Add keys for table Klienci

ALTER TABLE Klienci ADD CONSTRAINT Unique_Identifier4 PRIMARY KEY
(ID_klienta)
/

-- Table Uslugi

CREATE TABLE Uslugi(
    ID_uslugi Integer NOT NULL,
    Rodzaj_uslugi Varchar2(20 ) NOT NULL,
    Cena Float(126) NOT NULL,
    Data_realizacji Date NOT NULL,
    Pochodzenie_pradu Varchar2(30 ),
    ID_Dostawcy Integer NOT NULL
)
/

-- Create indexes for table Uslugi

CREATE INDEX IX_Oferuje ON Uslugi (ID_Dostawcy)
/

-- Add keys for table Uslugi

ALTER TABLE Uslugi ADD CONSTRAINT Unique_Identifier5 PRIMARY KEY (ID_uslugi)
/

-- Table Infrastruktury

```

```

CREATE TABLE Infrastruktury(
    ID_elementu_infrastruktury Integer NOT NULL,
    Typ_elementu_elektrycznego Varchar2(40 ) NOT NULL,
    Data_produkcji Date NOT NULL,
    Data_ostatniego_serwisu Date NOT NULL,
    Producent Varchar2(40 ) NOT NULL,
    Gwarancja Char(1 ) NOT NULL,
    ID_Dostawcy Integer NOT NULL,
    ID_Adresu Integer NOT NULL
)
/

-- Create indexes for table Infrastruktury

CREATE INDEX IX_Buduje ON Infrastruktury (ID_Dostawcy)
/

CREATE INDEX IX_Relationship6 ON Infrastruktury (ID_Adresu)
/

-- Add keys for table Infrastruktury

ALTER TABLE Infrastruktury ADD CONSTRAINT Unique_Identifier6 PRIMARY KEY
(ID_elementu_infrastruktury)
/

-- Table Inzynierowie

CREATE TABLE Inzynierowie(
    Prawo_jazdy Char(1 ) NOT NULL,
    Specjalizacja Varchar2(40 ) NOT NULL,
    ID_pracownika Integer NOT NULL
)
/

-- Add keys for table Inzynierowie

ALTER TABLE Inzynierowie ADD CONSTRAINT Unique_Identifier9 PRIMARY KEY
(ID_pracownika)
/

-- Table Konsultanci

CREATE TABLE Konsultanci(
    Ilosc_znanych_jezykow Integer NOT NULL,
    Mozliwosc_pracy_zdalnej Char(1 ) NOT NULL,
    ID_pracownika Integer NOT NULL
)
/

```

```

-- Add keys for table Konsultanci

ALTER TABLE Konsultanci ADD CONSTRAINT Unique_Identifier10 PRIMARY KEY
(ID_pracownika)
/

-- Table Firmy

CREATE TABLE Firmy(
    Nazwa przedsiebiorstwa Varchar2(600 ) NOT NULL,
    NIP Varchar2(13 ) NOT NULL,
    KRS Varchar2(10 ) NOT NULL,
    REGON Varchar2(14 ) NOT NULL,
    Przedstawiciel Varchar2(40 ) NOT NULL,
    ID klienta Integer NOT NULL
)
/

-- Add keys for table Firmy

ALTER TABLE Firmy ADD CONSTRAINT Unique_Identifier11 PRIMARY KEY (ID
klienta)
/

-- Table Osoba_prywatna

CREATE TABLE Osoba_prywatna(
    Typ_budynku Varchar2(40 ) NOT NULL,
    Karta_duzej_rodziny Char(1 ) NOT NULL,
    ID_klienta Integer NOT NULL
)
/

-- Add keys for table Osoba_prywatna

ALTER TABLE Osoba_prywatna ADD CONSTRAINT Unique_Identifier13 PRIMARY KEY
(ID_klienta)
/

-- Table Oddzial_Usluga

CREATE TABLE Oddzial_Usluga(
    ID_oddzialu Integer NOT NULL,
    ID_uslugi Integer NOT NULL
)
/

-- Table Oddzial_Klient

CREATE TABLE Oddzial_Klient(
    ID_oddzialu Integer NOT NULL,

```

```

        ID_klienta Integer NOT NULL
    )
    /

-- Table Oddzial_Pracownik

CREATE TABLE Oddzial_Pracownik(
    ID_oddzialu Integer NOT NULL,
    ID_pracownika Integer NOT NULL
)
/

-- Table Oddzial_Infrastruktura

CREATE TABLE Oddzial_Infrastruktura(
    ID_oddzialu Integer NOT NULL,
    ID_elementu_infrastruktury Integer NOT NULL
)
/

-- Table Pracownik_Usluga

CREATE TABLE Pracownik_Usluga(
    ID_pracownika Integer NOT NULL,
    ID_uslugi Integer NOT NULL
)
/

-- Table Adresy

CREATE TABLE Adresy(
    ID_Adresu Integer NOT NULL,
    Miasto Varchar2(20 ) NOT NULL,
    Ulica Varchar2(30 ) NOT NULL,
    Nr_lokalu Varchar2(5 ) NOT NULL,
    ID_Poczty Integer NOT NULL
)
/

-- Create indexes for table Adresy

CREATE INDEX IX_Relationship10 ON Adresy (ID_Poczty)
/

-- Add keys for table Adresy

ALTER TABLE Adresy ADD CONSTRAINT PK_Adresy PRIMARY KEY (ID_Adresu)
/

-- Table and Columns comments section

```



```

COMMENT ON COLUMN Adresy.ID_Adresu IS 'Unikatowy klucz główny w relacji
adresy'
/
COMMENT ON COLUMN Adresy.Ulica IS 'Ulica'
/
COMMENT ON COLUMN Adresy.Nr_lokalu IS 'Numer lokalu'
/

-- Table Wlasciciele

CREATE TABLE Wlasciciele(
    ID_Wlasciciela Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    ID_Dostawcy Integer NOT NULL
)
/

-- Create indexes for table Wlasciciele

CREATE INDEX IX_Relationship8 ON Wlasciciele (ID_Dostawcy)
/

-- Add keys for table Wlasciciele

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

-- Table and Columns comments section

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

-- Table Kierownicy

CREATE TABLE Kierownicy(
    ID_Kierownika Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    ID_oddzialu Integer NOT NULL
)
/

-- Create indexes for table Kierownicy

```

```

CREATE INDEX IX_Relationship9 ON Kierownicy (ID_oddzialu)
/

-- Add keys for table Kierownicy

ALTER TABLE Kierownicy ADD CONSTRAINT PK_Kierownicy PRIMARY KEY
(ID_Kierownika)
/

-- Table and Columns comments section

COMMENT ON COLUMN Kierownicy.ID_Kierownika IS 'Klucz identyfikujący
kierownika'
/
COMMENT ON COLUMN Kierownicy.Imie IS 'Imie kierownika'
/
COMMENT ON COLUMN Kierownicy.Nazwisko IS 'Nazwisko kierownika'
/

-- Table Poczty

CREATE TABLE Poczty(
    ID_Poczty Integer NOT NULL,
    Poczta Varchar2(20 ) NOT NULL,
    Kod_pocztowy Varchar2(6 ) NOT NULL
)
/

-- Add keys for table Poczty

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

-- Table and Columns comments section

COMMENT ON COLUMN Poczty.ID_Poczty IS 'Klucz identyfikujący pocztę'
/
COMMENT ON COLUMN Poczty.Poczta IS 'Poczta'
/
COMMENT ON COLUMN Poczty.Kod_pocztowy IS 'Kod pocztowy'
/

-- Table Stanowiska

CREATE TABLE Stanowiska(
    ID_Stalowiska Integer NOT NULL,
    Nazwa Varchar2(20 ) NOT NULL,
    Opis Varchar2(400 ) NOT NULL
)
/

```

```

-- Add keys for table Stanowiska

ALTER TABLE Stanowiska ADD CONSTRAINT PK_Stalowiska PRIMARY KEY
(ID_Stalowiska)
/

ALTER TABLE Stanowiska ADD CONSTRAINT Nazwa UNIQUE (Nazwa)
/

-- Table and Columns comments section

COMMENT ON COLUMN Stanowiska.ID_Stalowiska IS 'Unikatowy identyfikator
stawowiska'
/
COMMENT ON COLUMN Stanowiska.Nazwa IS 'Nazwa stawowiska'
/
COMMENT ON COLUMN Stanowiska.Opis IS 'Opis stawowiska'
/

-- Table Wynagrodzenia

CREATE TABLE Wynagrodzenia(
  ID_Wynagrodzenia Integer NOT NULL,
  Pensja_pod Number(8,2) NOT NULL,
  Pensja_dod Number(8,2),
  Data Date NOT NULL,
  ID_pracownika Integer NOT NULL
)
/

-- Create indexes for table Wynagrodzenia

CREATE INDEX IX_Relationship31 ON Wynagrodzenia (ID_pracownika)
/

-- Add keys for table Wynagrodzenia

ALTER TABLE Wynagrodzenia ADD CONSTRAINT PK_Wynagrodzenia PRIMARY KEY
(ID_Wynagrodzenia)
/

-- Table and Columns comments section

COMMENT ON COLUMN Wynagrodzenia.ID_Wynagrodzenia IS 'Unikatowy identyfikator
wynagrodzenia'
/
COMMENT ON COLUMN Wynagrodzenia.Pensja_pod IS 'Pensja podstawowa pracownika'
/
COMMENT ON COLUMN Wynagrodzenia.Pensja_dod IS 'Pensja dodatkowa'
/
COMMENT ON COLUMN Wynagrodzenia.Data IS 'Data wypłaty wynagrodzenia'

```

/

-- Create foreign keys (relationships) section

```
ALTER TABLE Oddzialy ADD CONSTRAINT Otwiera FOREIGN KEY (ID_Dostawcy)
REFERENCES Dostawcy (ID_Dostawcy)
```

/

```
ALTER TABLE Infrastruktury ADD CONSTRAINT Buduje FOREIGN KEY (ID_Dostawcy)
REFERENCES Dostawcy (ID_Dostawcy)
```

/

```
ALTER TABLE Uslugi ADD CONSTRAINT Oferuje FOREIGN KEY (ID_Dostawcy)
REFERENCES Dostawcy (ID_Dostawcy)
```

/

```
ALTER TABLE Dostawcy ADD CONSTRAINT Biuro_ma_adres FOREIGN KEY (ID_Adresu)
REFERENCES Adresy (ID_Adresu)
```

/

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_ma_adres_zamieszkania
FOREIGN KEY (ID_Adresu) REFERENCES Adresy (ID_Adresu)
```

/

```
ALTER TABLE Klienci ADD CONSTRAINT Klient_ma_adres FOREIGN KEY (ID_Adresu)
REFERENCES Adresy (ID_Adresu)
```

/

```
ALTER TABLE Oddzialy ADD CONSTRAINT Oddzial_ma_adres FOREIGN KEY (ID_Adresu)
REFERENCES Adresy (ID_Adresu)
```

/

```
ALTER TABLE Infrastruktury ADD CONSTRAINT Infrastruktura_ma_swoje_polozenie
FOREIGN KEY (ID_Adresu) REFERENCES Adresy (ID_Adresu)
```

/

```
ALTER TABLE Wlasciciele ADD CONSTRAINT Biuro_ma_wlasciciela FOREIGN KEY  
(ID_Dostawcy) REFERENCES Dostawcy (ID_Dostawcy)  
/
```

```
ALTER TABLE Kierownicy ADD CONSTRAINT Oddzial_ma_kierownika FOREIGN KEY  
(ID_oddzialu) REFERENCES Oddzialy (ID_oddzialu)  
/
```

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

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_ma_stanowisko FOREIGN KEY  
(ID_Stanowiska) REFERENCES Stanowiska (ID_Stanowiska)  
/
```

```
ALTER TABLE Wynagrodzenia ADD CONSTRAINT Otrzymuje_wynagrodzenia FOREIGN KEY  
(ID_pracownika) REFERENCES Pracownicy (ID_pracownika)  
/
```

5.4 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

- Pozyskanie numeru id oddziału i daty założenia dla daty założenia mniejszej od 20/01/01

Worksheet	Query Builder														
	<pre>SELECT ID_ODDZIALU, DATA_ZALOZENIA FROM ODDZIALY WHERE data_zalozenia < '20/01/01'</pre>														
	<table><tr><th>ID_ODDZIALU</th><th>DATA_ZALOZENIA</th></tr><tr><td>1</td><td>19/12/20</td></tr><tr><td>2</td><td>05/06/23</td></tr><tr><td>3</td><td>02/08/21</td></tr><tr><td>4</td><td>18/10/26</td></tr><tr><td>5</td><td>08/02/14</td></tr><tr><td>6</td><td>00/01/05</td></tr></table>	ID_ODDZIALU	DATA_ZALOZENIA	1	19/12/20	2	05/06/23	3	02/08/21	4	18/10/26	5	08/02/14	6	00/01/05
ID_ODDZIALU	DATA_ZALOZENIA														
1	19/12/20														
2	05/06/23														
3	02/08/21														
4	18/10/26														
5	08/02/14														
6	00/01/05														

- Pozyskanie imienia, nazwiska, wykształcenia, maila i daty urodzenia pracowników płci męskiej.

Worksheet

Query Builder

```
SELECT IMIE, NAZWISKO, WYKSZTALCENIE, EMAIL, DATA_URODZENIA
FROM PRACOWNICY
WHERE PLEC = 'M'
```

	IMIE	NAZWISKO	WYKSZTALCENIE	EMAIL	DATA_URODZENIA
1	Michał	Wilak	Srednie	Michalak@wp.pl	90/06/13
2	Tomasz	Waleczny	Wyzsze	Walczak@o2.pl	00/11/25
3	Adam	Laziuk	Podstawowe	adamlaziuk@gmail.com	70/10/21

Bibliografia

Przy projektowaniu naszej bazy danych posłużyliśmy się następującymi źródłami:

<https://www.sqlpedia.pl/projektowanie-i-normalizacja-bazy-danych/>
