

Grzegorz Wszola, Patryk Jankowicz

Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych

Projekt BADA cz. 1, grupa dziekańska: 2 - Cyberbezpieczeństwo

12 grudnia 2022

Spis treści

1. Zakres i cel projektu	2
2. Definicja systemu	2
2.1. Perspektywa właściciela	2
2.2. Perspektywa administratora	2
2.3. Perspektywa pracownika	2
2.4. Perspektywa klienta	2
3. Model konceptualny	2
3.1. Definicja zbiorów encji określonych w projekcie	2
3.2. Ustalenie związków między encjami i ich typów	3
3.3. Określenie atrybutów i ich dziedzin	4
3.4. Dodatkowe reguły integralnościowe	5
3.5. Klucze kandydujące i główne	5
3.6. Schemat ER na poziomie konceptualnym	6
3.7. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady	6
4. Model logiczny	6
4.1. Charakterystyka modelu relacyjnego	6
4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady	7
4.2.1. Usunięcie związków "wielu do wielu"	7
4.2.2. Dodanie kluczy obcych	7
4.2.3. Zmiana nazw	7
4.3. Proces normalizacji – analiza i przykłady	7
4.3.1. Pierwsza postać normalna	7
4.3.2. Druga postać normalna	8
4.3.3. Trzecia postać normalna	8
4.4. Schemat ER na poziomie modelu logicznego	8
4.5. Więzy integralności	8
4.6. Proces denormalizacji – analiza i przykłady	9
5. Faza fizyczna	9
5.1. Projekt transakcji i weryfikacja ich wykonalności	9
5.2. Strojanie bazy danych – dobór indeksów	9
5.3. Skrypt SQL zakładający bazę danych	10
5.4. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych	24

1. Zakres i cel projektu

Celem projektu było zaprojektowanie bazy danych poprzez: opracowanie modeli konceptualnego i logicznego, a finalnie zaimplementowanie ich do bazy danych Oracle z wykorzystaniem SQL. Naszym tematem był: Operator sieci komórkowej. W projekcie zawarliśmy elementy "techniczne" tj. infrastruktura używana przez operatorów oraz te dostępne dla każdego, np. punkty sprzedaży, nieruchomości... Modele konceptualny, a potem logiczny wykonaliśmy z użyciem programu Toad data modeler - w nim także wygenerowaliśmy skrypt SQL. Natomiast finalną implementację przeprowadziliśmy w SQL Developer i VPN, korzystając z udostępnionych zasobów politechnicznych.

2. Definicja systemu

Baza danych o operatorze sieci komórkowej powinna udostępnia różnym użytkownikom różne możliwości zarówno pod kątem uprawnień dostępu do danych, jak i możliwości ich edytowania. Role jak definiujemy dla naszej bazy danych to: właściciel, administrator, pracownik, klient.

2.1. Perspektywa właściciela

Właściciel jest osobą której firma jest własnością. W związku z tym ma wgląd do wszystkich danych i może je edytować.

2.2. Perspektywa administratora

Administrator zawiaduje jakąś jednostką administracyjną. Może być to jakaś nieruchomość np.: biuro, czy punkt sprzedaży, albo infrastruktura np.: serwerownia, czy agregatornia. Ma on dostęp do danych technicznych dotyczących jednostki, której jest administratorem i danych o pracownikach, którzy w tej jednostce pracują. Dane do których ma dostęp, może również modyfikować.

2.3. Perspektywa pracownika

Pracownik jest osobą, która pracuje w danej jednostce. Ma on dostęp do danych dotyczących jednostki w której pracuje, pod warunkiem, że potrzebuje ich do wykonywania swojej pracy i dostęp do danych klientów pod identycznym warunkiem. Może je modyfikować w ograniczonym zakresie.

2.4. Perspektywa klienta

Klient to osoba korzystająca z usług oferowanych przez operatora sieci komórkowej. Ma on tylko wgląd do danych opisujących go i usługi z których skorzystał lub korzysta. Nie ma możliwości ich samodzielnej modyfikacji.

3. Model konceptualny

3.1. Definicja zbiorów encji określonych w projekcie

1. **Operator sieci komórkowej** - główna encja wyjściowa reprezentująca tytułowy obiekt
2. **Nieruchomość** - encja reprezentująca obiekty 'budynków' należących do danego operatora. Np. biuro, w którym pracują pracownicy
3. **Klient** - encja reprezentuje fizyczną osobę - klienta korzystającego z usług operatora sieci komórkowej
4. **Infrastruktura** - encja reprezentująca "techniczne" zaplecze operatora, umożliwiające działanie sieci. Każda infrastruktura jest zarządzana przez administratora.
5. **Usługa** - encja reprezentuje działania, z których klient może skorzystać w ramach oferty operatora, np. konsultacja.
6. **Pracownik** - encja reprezentująca osobę fizyczną, pracującą dla danego operatora sieci komórkowej. Wśród nich mogą być m.in administratorzy odpowiedzialni za działanie infrastruktury jak i zarządzanie nieruchomościami.

3.2. Ustalenie związków między encjami i ich typów

Encja	Lewa krotność	Związek	Encja	Prawa krotność
Operator sieci komórkowej	1:1	zatrudnia	Pracownik	0:N

Operator zatrudnia 0 lub wielu pracowników, natomiast każdy pracownik, może (a nawet musi) być zatrudniony tylko przez jednego operatora.

Encja	Lewa krotność	Związek	Encja	Prawa krotność
Operator sieci komórkowej	1:1	posiada klienta	Klient	0:N

Operator może posiadać 0 lub wielu klientów, ale klient może "być klientem" tylko jednego operatora.

Encja	Lewa krotność	Związek	Encja	Prawa krotność
Operator sieci komórkowej	1:1	Oferuje	Usługa	0:N

Operator oferuje 0 lub wiele usług, jednakże dana usługa może być oferowana tylko przez jednego operatora.

Encja	Lewa krotność	Związek	Encja	Prawa krotność
Operator sieci komórkowej	1:1	Dysponuje	Infrastruktura	0:N

Operator dysponuje żadnym elementem lub wieloma elementami infrastruktury, ale infrastruktura może być wykorzystywana tylko przez 1 operatora.

Encja	Lewa krotność	Związek	Encja	Prawa krotność
Operator sieci komórkowej	1:1	posiada nieruchomość	Nieruchomość	0:N

Operator posiada 0 lub wiele nieruchomości, ale nieruchomość może być używana tylko przez jednego operatora.

Encja	Lewa krotność	Związek	Encja	Prawa krotność
Pracownik	1:M	Wykonuje	Usługa	0:N

Pracownik wykonuje 0 lub wiele usług, a usługa musi być wykonywana przez co najmniej jednego lub więcej pracowników.

Encja	Lewa krotność	Związek	Encja	Prawa krotność
Klient	1:1	Korzysta	Usługa	0:N

Klient korzysta z 0 lub wielu usług oferowanych przez operatora, ale dana usługa może być zarejestrowana/wykorzystana tylko na/przez jednego klienta.

Encja	Lewa krotność	Związek	Encja	Prawa krotność
Pracownik	1:M	Obsługuje	Infrastruktura	0:N

Pracownik obsługuje 0 lub wiele infrastruktur - nie każdy pracownik np. Sprzedawca musi być odpowiedzialny za infrastrukturę. Infrastruktura natomiast musi być obsługiwana przez co najmniej jednego lub wielu pracowników.

Encja	Lewa krotność	Związek	Encja	Krotność
Pracownik	1:M	Pracuje w	Nieruchomość	0:N

Pracownik może, ale nie musi (np. Technik) pracować w nieruchomości - np. biurze, ale w każdej nieruchomości musi pracować przynajmniej jeden lub wielu pracowników.

3.3. Określenie atrybutów i ich dziedziny

Encja	Nazwa	Domena	Typ danych	Parametr	PK	Oblowidkowe	Opis
Operator_sieci_komorkowej	Id_operatora_sieci_komorkowej		Integer		TRUE		
Operator_sieci_komorkowej	Nazwa_operatora		VarChar(%pl)	100	FALSE	TRUE	
Operator_sieci_komorkowej	Wlasciciel		VarChar(%pl)	200	FALSE	TRUE	Imie i nazwisko wlasciciela.
Operator_sieci_komorkowej	Data_zalozenia		Date		FALSE	TRUE	
Pracownik	Kapital		Money		TRUE		
Pracownik	Id_pracownika		Integer		TRUE	TRUE	
Pracownik	Imie		VarChar(%pl)	20	FALSE	TRUE	
Pracownik	Nazwisko		VarChar(%pl)	30	FALSE	TRUE	
Pracownik	PESEL		Character(%pl)	11	FALSE	FALSE	Pole nieobowiazkowe; jedenastoznakowy numer PESEL
Pracownik	Data_urodzenia		Date		FALSE	TRUE	
Pracownik	Data_zatrudnienia		Date		FALSE	TRUE	
Pracownik	Rodzaj_wyksztalcenia	wyksztalcenieD			FALSE	TRUE	Wybor wykształcenia ze zbioru: podstawowe, srednie, wyzsze
Pracownik	Pensja		Money		FALSE	TRUE	Pensja; maksymalna wartosc 32767 zł.
Pracownik	Nr_telefonu		VarChar(%pl)	9	FALSE	FALSE	Numer telefon - max. 9 znakow - bez spacji
Pracownik	Adres_email		VarChar(%pl)	50	FALSE	FALSE	Adres email - max. 50 znakow
Pracownik	Adres		VarChar(%pl)	400	FALSE	TRUE	Adres zamieszkania pracownika - 400 znakow maksymalnie
Pracownik	Czy_mobilnie		Boolean		FALSE	TRUE	boolean - czy pracownik pracuje w biurze, czy w domu lub w terenie
Pracownik	Rodzaj_stanowiska	StanowiskoD			FALSE	TRUE	
Klient	Id_klienta		SmallInt		TRUE	TRUE	
Klient	Imie		VarChar(%pl)	20	FALSE	TRUE	
Klient	Nazwisko		VarChar(%pl)	30	FALSE	TRUE	
Klient	Rodzaj_pakietu	pakietD		100	FALSE	FALSE	
Klient	Numer_telefonu		VarChar(%pl)	9	FALSE	FALSE	
Klient	Adres_email		VarChar(%pl)	400	FALSE	TRUE	
Klient	Czy_newsletter		Boolean		FALSE	TRUE	Flaga; czy wysylac newsletter na maila?
Klient	Adres		VarChar(%pl)	400	FALSE	TRUE	
Usługa	Id_uslugi		Integer		TRUE	TRUE	
Usługa	Rodzaj_uslugi	UsługaD			FALSE	TRUE	Wybór jednej z listy możliwych uslug - Wsparcie_tekniczne,Serwis,Sprzedaz,Reklamacja,Konsultacja
Usługa	Koszt		Money		FALSE	TRUE	Cena za wybrane uslugi
Usługa	Czas_trwania		VarChar(%pl)	3	FALSE	FALSE	Nieobowiazkowe pole, czas trwania uslugi podany w minutach
Usługa	Czy_zdalna		Boolean		FALSE	TRUE	Boolean - czy usługa jest realizowana zdalnie, np. konsultacja przez telefon
Usługa	Czy_gwarancja		Boolean		FALSE	TRUE	Boolean - czy usługa wykonywana jest w ramach gwarancji
Nieruchomosc	Id_nieruchomosci		Integer		TRUE	TRUE	
Nieruchomosc	Adres		VarChar(%pl)	400	FALSE	TRUE	
Nieruchomosc	Rodzaj_nieruchomosci		VarChar(%pl)	30	FALSE	TRUE	Wybór_nieruchomosci z: Biuro, Punkt_sprzedazy, Magazyn
Nieruchomosc	Powierzchnia		Integer		FALSE	TRUE	Powierzchnia nieruchomosci wyrazona w m2
Nieruchomosc	Waznosc_badan_ppoz		Date		FALSE	TRUE	Data do kiedy wazna sa badania przeciw pozarowe - np. waznosc gasnic
Nieruchomosc	Nazwa_firmy_ochroniarskiej		VarChar(%pl)	100	FALSE	FALSE	Nazwa firmy ochroniarskiej
Nieruchomosc	Maksymalna_liczba_stanowisk		SmallInt		FALSE	FALSE	
Nieruchomosc	Czy_monitorowane		Boolean		FALSE	TRUE	boolean - czy obiekt monitorowany
Infrastruktura	Id_infrastruktury		Integer		TRUE	TRUE	
Infrastruktura	Adres		VarChar(%pl)	400	FALSE	TRUE	boolean - czy obiekt jest monitorowany
Infrastruktura	Rodzaj_infrastruktury	InfrastrukturaD		30	FALSE	TRUE	
Infrastruktura	Administrator		VarChar(%pl)	150	FALSE	FALSE	boolean - czy infrastruktura jest krytyczna, niezbedna do dzialania sieci
Infrastruktura	Czy_krytyczna		Boolean		FALSE	TRUE	Wartosc infrastruktury - przydane np. w sytuacji kradziezy czy uszkodzenia
Infrastruktura	Wartosc		Money		FALSE	TRUE	Data kiedy ostatnio zostala przeprowadzona konserwacja
Infrastruktura	Data_konserwacji		Date		FALSE	TRUE	boolean - czy obiekt monitorowany
Infrastruktura	Czy_monitorowane		Boolean		FALSE	TRUE	Nazwa firmy ochroniarskiej - maks. 100 znakow
Infrastruktura	Nazwa_firmy_ochroniarskiej		VarChar(%pl)	100	FALSE	FALSE	Czy infrastruktura jest ma zasilanie awaryjne
Infrastruktura	Czy_zasilanie_awaryjne		Boolean		FALSE	TRUE	

Nazwa	Typ danych	Parametr	Zakres dziedziny
wykształcenieD	VarChar(%p1)	30	Podstawowe, Srednie, Wyzsze
pakietD	VarChar(%p1)	50	Nielimitowane_rozmowy, Nielimitowane_SMSy, Nielimitowany_internet, Nielimitowany, Nielimitowane_SMSy_i_rozmowy
UsługaD	VarChar(%p1)	30	Wsparcie_techiczne, Serwis, Sprzedaz, Reklamacja, Konsultacja
NieruchomoscD	VarChar(%p1)	25	Biuro, Punkt_sprzedazy, Magazyn
InfrastrukturaD	VarChar(%p1)	50	Stacja_przekaznikowa, Serwerownia, Agregatornia
StanowiskoD	VarChar(%p1)	25	Administrator, Sprzedawca, Serwisant, Konsultant, Technik, Wozny

3.4. Dodatkowe reguły integralnościowe

Zasadami, narzuconymi przez model relacyjny są między innymi:

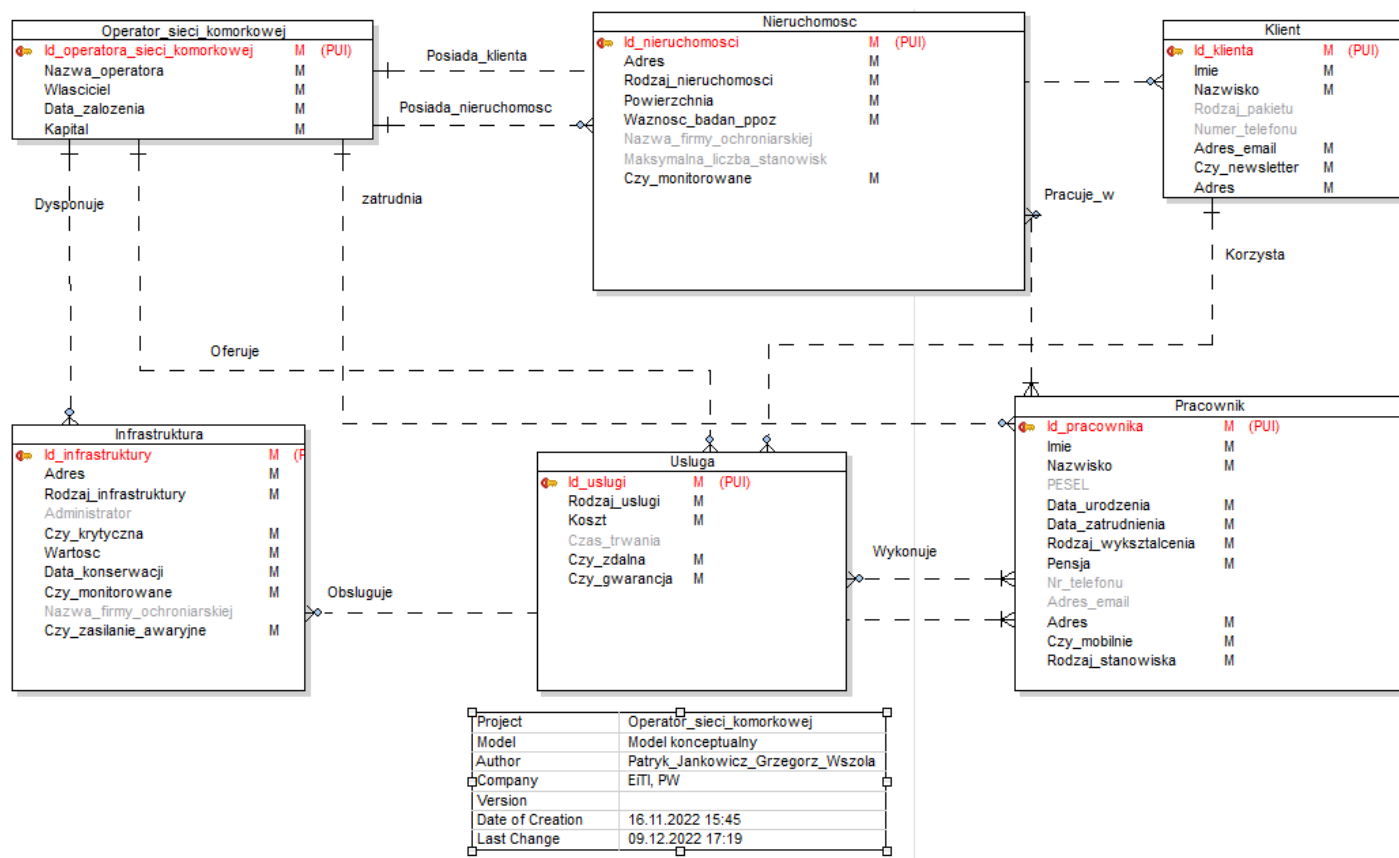
- Spełnienie zdefiniowanych i nałożonych ograniczeń przez wartości danych. Dla każdego atrybutu staraliśmy się dobrać odpowiednie dziedziny, typy danych. Dodatkowo na etapie implementacji upewniliśmy się, że wszystkie wstawiane wartości do bazy są poprawne.
- Każda tabela posiada klucz główny, a jego wartości w ramach tabeli są unikatowe i różne od wartości NULL. W przypadku toad data modelera, przy zaznaczeniu opcji klucz główny automatycznie zaznaczała się opcja NOT NULL, bez możliwości zmiany.
- każda wartość klucza obcego jest równa jakiejś wartości klucza głównego w powiązanej tabeli nad rzędną lub ewentualnie jest równa NULL. W naszym przypadku klucz obcy jest zawsze sztucznym kluczem głównym innej encji, co spełnia założenie.

3.5. Klucze kandydujące i główne

Encja	Nazwa atrybutu	Klucz główny
Operator_sieci_komorkowej	Id_operatora_sieci_komorkowej	TRUE
Pracownik	Id_pracownika	TRUE
Klient	Id_klienta	TRUE
Usługa	Id_uslugi	TRUE
Nieruchomosc	Id_nieruchomosci	TRUE
Infrastruktura	Id_infrastruktury	TRUE

Dla każdej encji postanowiliśmy utworzyć sztuczny klucz główny, którego nazwa to Id wraz z odpowiadającą nazwą. Dzięki temu unikneliśmy problemy np. z kluczami wielowartościowymi oraz uprościliśmy sobie proces normalizacji. Dodatkowo dla każdego z nich stworzyliśmy odpowiednio własną sekwencję. Kluczem kandydującym mogłby być np. numer pesel w przypadku pracownika, jednak zakładamy że pracownicy nie muszą mieć polskiego obywatelstwa, więc nie można wykorzystać w ten sposób tego atrybutu.

3.6. Schemat ER na poziomie konceptualnym



Rys. 1. Schemat modelu konceptualnego

3.7. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

- **Pułapki wachlarzowe** pojawiają się, w sytuacji kiedy zależności między encjami nie są jednoznaczne.
- **Pułapki szczelinowe** pojawiają się, w sytuacji kiedy model sugeruje istnienie powiązania pomiędzy zbiorami encji, którego w rzeczywistości nie ma.

Dzięki dokładnemu przemyśleniu zastosowanych związków, wydaje nam się że nasz projekt nie ma żadnych z wyżej wymienionych pułapek.

4. Model logiczny

4.1. Charakterystyka modelu relacyjnego

Model relacyjny bazuje na relacjach czyli encjach opisujących jakiś fragment rzeczywistości. Relacje są opisane za pomocą atrybutów, a pomiędzy różnymi relacjami zachodzą różnego rodzaju związki. Dzięki temu osiągnięta jest duża niezależność danych. Same relacje nazywają się odmiennie od siebie i wszystkie mają atrybuty, które również różnią się od siebie nazwami. Każdy atrybut może przyjmować tylko wartości ze swojej dziedziny, a każda komórka powinna być polem atomowym. Ponadto nie może dochodzić do powtarzania się krotek. By osiągnąć te założenia przeprowadziliśmy różne procesy dokładnie niżej opisane.

4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

Tworzenie modelu logicznego rozpoczęliśmy od konwersji modelu conceptualnego za pomocą konwertera wbudowanego w aplikację Data Toad Modeler, w której utworzyliśmy model conceptualny.

4.2.1. Usunięcie związków "wielu do wielu"

Największą zmianą, osiągniętą za pomocą konwertera było usunięcie związków encji typu "wielu do wielu" i stworzenie dodatkowych encji w miejscu tych związków. Tabel łączących zawierające klucze główne łączonych encji. W tabeli są kolejno: nazwa pierwszej encji, nazwa drugiej encji, nazwa nowoutworzonej encji, krotność związku nowej encji z 1 encją, krotność związku nowej encji z 2 encją.

Encja 1	Encja 2	Nowe encja	Związek z 1 encją	Związek z 2 encją
Pracownicy	Usługi	Pracownicy-Usługi	1..1 to 0..n	1..1 to 1..n
Pracownicy	Infrastruktury	Pracownicy-Infrastruktury	1..1 to 0..n	1..1 to 1..n
Pracownicy	Nieruchomości	Pracownicy-Nieruchomości	1..1 to 0..n	1..1 to 1..n

4.2.2. Dodanie kluczy obcych

Ponadto konwerter w miejscach gdzie encje były w jakimś związku dodawał klucze obce. W tabeli kolejno: Nazwa encji, nazwy kluczy obcych w danej encji.

Nazwa encji	Kucze obce
Właściciele	Id-operatora-sieci-komórkowej, Id-adresu
Klienci	Id-operatora-sieci-komórkowej, Id-adresu
Adresy	Id-poczty
Nieruchomości	Id-operatora-sieci-komórkowej, Id-adresu, Id-firmy-ochroniarskiej
Pakiety-usług	Id-klienta
Pracownicy	Id-operatora-sieci-komórkowej, Id-adresu, Id-stanowiska
Usługi	Id-operatora-sieci-komórkowej, Id-klienta, Id-Listy-usług
Firmy ochroniarskie	Id-adresu
Infrastruktury	Id-operatora-sieci-komórkowej, Id-adresu, Id-firmy-ochroniarskiej

4.2.3. Zmiana nazw

Kolejnym etapem była zmiana nazw z liczb pojedynczych na liczby mnogie. W tabeli kolejno: nazwa encji przed zmianą, nazwa encji po zmianie.

Nazwa encji przed zmianą	Nazwa encji po zmianie
Operator sieci komórkowej	Operatorzy sieci komórkowych
Nieruchomość	Nieruchomości
Klient	Klienci
Usługa	Usługi
Pracownik	Pracownicy
Infrastruktura	Infrastruktury

4.3. Proces normalizacji – analiza i przykłady

4.3.1. Pierwsza postać normalna

By doprowadzić bazę do pierwszej postaci normalnej musieliśmy zadbać by atrybuty wszystkich relacji były atrybutami atomowymi i by nie istniały żadne powtarzające się grupy. Realizując to stworzyliśmy nową encję - "Adresy" gdzie znajdowały się wszystkie atrybuty adresów, takie jak miejscowość, ulica, numer budynku, numer lokalu i poczta. Jako iż zostawienie poczty w takiej postaci mogło tworzyć grupy to postanowiliśmy utworzyć kolejną encję - "Poczty" z atrybutami takimi jak numer poczty i miejscowość poczty. Innym przykładem normalizacji jest rozbicie pola nieatomowego właściciela w encji "Operatorzy sieci komórkowej". Zamiast tego stworzyliśmy nową encję - "Właściciele" która posiada takie atrybuty jak imię, nazwisko i numer telefonu. Innym przykładem likwidacji możliwości powstawania grup jest utworzenie encji "Listy usług" zawierające atrybuty takie jak rodzaj usługi, koszt usługi i informację, czy usługa jest wykonywana zdalnie i czy jest gwarancja po wykonaniu usługi. Również w ramach normalizacji do pierwszej postaci normalnej utworzyliśmy encję "Firmy ochroniarskie", by pozbyć się pola nieatomowego o tej samej

nazwie w encjach "Nieruchomości" i "Infrastruktury". Zawiera ona atrybuty takie jak nazwa, awaryjny numer telefonu i informację czy oferuje monitoring.

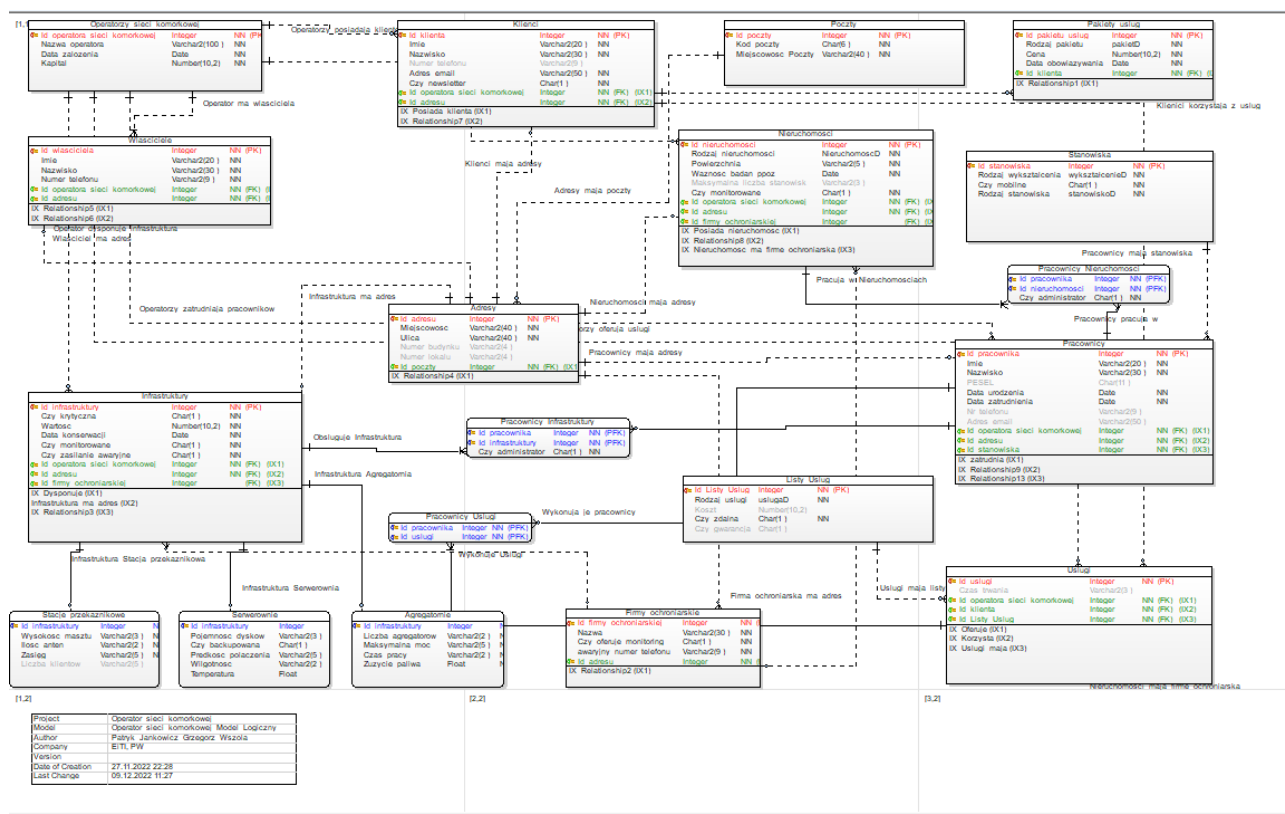
4.3.2. Druga postać normalna

Postać drugą normalną można uzyskać kiedy baza ma pierwszą postać normalną i gdy żadne atrybuty niebędące kluczami kandydującymi lub nie wchodzące w skład klucza kandydującego są w pełni funkcyjnie zależne od klucza głównego oraz atrybuty te zależą od całości klucza, a nie od jego części. W momencie w którym wszystkie instniejące klucze są kluczami prostymi i baza jest w pierwszej postaci normalnej to jest ona od razu również w drugiej postaci normalnej. Po przeprowadzeniu analizy naszego projektu stwierdziliśmy, że nas dotyczy przypadek drugi, gdyż wszystkie klucze w projekcie są kluczami sztucznymi, prostymi. Stąd też nie wykonywaliśmy żadnych działań w kontekście doprowadzenia bazy do postaci 2NF.

4.3.3. Trzecia postać normalna

W przypadku trzeciej postaci normalnej muszą być spełnione dwa warunki: musi być ona w drugiej postaci normalnej i wszystkie atrybuty w tych relacjach muszą zależeć tylko i wyłącznie od klucza głównego. W tym celu utworzyliśmy encję "Stanowiska", w której zawarliśmy wszystkie atrybuty, które wynikały z radzaju stanowiska, a nie zależały od klucza głównego encji "Pracownicy". Tymi atrybutami są rodzaj stanowiska, rodzaj wykształcenia i informacja czy stanowisko jest mobilne. Po wykonaniu tych zmian baza danych została doprowadzona do trzeciej postaci normalnej.

4.4. Schemat ER na poziomie modelu logicznego



Rys. 2. Schemat modelu logicznego - większy na załączonej kartce

4.5. Więzy integralności

Integralność danych służy temu, żeby wszystkie dane były ze sobą spójne i by po jakichś modyfikacjach, czy to aktualizacji, dodawaniu, czy usuwaniu danych nadal wszystkie związki były prawdziwe. Integralność danych osiągamy za pomocą prawidłowo zaprojektowanego systemu kluczy głównych. W naszym projekcie wszystkie klucze główne są

polami unikatowymi i obowiązkowymi. Nie jest dopuszczona sytuacja, by dwie krotki miały dokładnie ten sam klucz główny, ani by jakieś krotki nie posiadały swojego klucza głównego. Dzięki temu można tworzyć związki pomiędzy relacjami za pomocą kluczy obcych.

4.6. Proces denormalizacji – analiza i przykłady

Denormalizacja jest procesem polegającym na cofaniu zmian wprowadzonych w trakcie przeprowadzania normalizacji. Denormalizację przeprowadza się w celu zwiększenia wydajności systemu po przez na przykład umożliwienie powtarzania się grup, czy łączenie tabel w związkach typu "jeden do jeden". Osiąga się to często kosztem elastyczności systemu i zwiększeniem skomplikowania projektu. Po analizie naszego projektu nie znaleźliśmy miejsc w których należało by przeprowadzić denormalizację, gdyż uznaliśmy, że nasza baza danych nie jest dużym projektem i system będzie w stanie spełnić wymagania w zakresie wydajności. Miejscami gdzie ewentualnie można by przeprowadzić denormalizację są encje "Listy usług" i "Usługi". Te encje można by połączyć tworząc jedną encję, co zapewne ułatwiłoby wyszukiwanie tych danych, ale doprowadzałoby do redundancji niektórych danych i powstawania powtarzających się grup.

5. Faza fizyczna

5.1. Projekt transakcji i weryfikacja ich wykonalności

- Dodawanie rekordów do bazy danych (załączony kod SQL) - **Wykonalne**
- Wypisanie dodanych rekordów (zrzuty ekranu w punkcie 5.4) - **Wykonalne**
- Modyfikacja dodanych rekordów (zrzuty ekranu w punkcie 5.4) - **Wykonalne**
- Usuwanie rekordu z bazy danych (zrzuty ekranu w punkcie 5.4) - **Wykonalne**

5.2. Strojenie bazy danych – dobór indeksów

1. Dla pracownicy:

```
CREATE INDEX IX_zatrudnia ON Pracownicy (Id_operatora_sieci_komorkowej)
CREATE INDEX IX_Relationship9 ON Pracownicy (Nr_adresu)
CREATE INDEX IX_Relationship13 ON Pracownicy (Id_stanowiska)
```

2. Dla klienci:

```
CREATE INDEX IX_Posiada_klienta ON Klienci (Id_operatora_sieci_komorkowej)
CREATE INDEX IX_Relationship7 ON Klienci (Nr_adresu)
```

3. Dla usługi:

```
CREATE INDEX IX_Oferuje ON Uslugi (Id_operatora_sieci_komorkowej)
CREATE INDEX IX_Korzysta ON Uslugi (Id_klienta)
CREATE INDEX IX_Uslugi_maja ON Uslugi (Id_Listy_Uslug)
```

4. Dla nieruchomości

```
CREATE INDEX IX_Posiada_nieruchomosc ON Nieruchomosci (Id_operatora_sieci_komorkowej)
CREATE INDEX IX_Relationship8 ON Nieruchomosci (Nr_adresu)
CREATE INDEX IX_Nieruchomosc_ma_firme_ochroniarska ON Nieruchomosci (Id_firmy_ochroniarskiej)
```

5. Dla infrastruktury

```
CREATE INDEX IX_Dysponuje ON Infrastruktury (Id_operatora_sieci_komorkowej)
CREATE INDEX Infrastruktura_ma_adres ON Infrastruktury (Nr_adresu)
CREATE INDEX IX_Relationship3 ON Infrastruktury (Id_firmy_ochroniarskiej)
```

6. Dla adresy

```
CREATE INDEX IX_Relationship4 ON Adresy (Id_poczty)
```

7. Dla właścicieli

```
CREATE INDEX IX_Relationship5 ON Wlasciciele (Id_operatora_sieci_komorkowej)
CREATE INDEX IX_Relationship6 ON Wlasciciele (Id_adresu)
```

8. Dla pakiety usług:

```
CREATE INDEX IX_Relationship1 ON Pakiety_uslug (Id_klienta)
```

9. Dla firmy ochroniarskie:

```
CREATE INDEX IX_Relationship2 ON Firmy_ochroniarskie (Id_adresu)
```

5.3. Skrypt SQL zakładający bazę danych

```
/*
Created: 27.11.2022
Modified: 09.12.2022
Project: Operator_sieci_komorkowej
Model: Operator_sieci_komorkowej_Model_Logiczny
Company: EITI, PW
Author: Patryk_Jankowicz_Grzegorz_Wszola
Database: Oracle 12c Release 2
*/

-- Create sequences section -----

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

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

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

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

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

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

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

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

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

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

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

```

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

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

-- Create tables section -----

-- Table Stacje_przekaznikowe

CREATE TABLE Stacje_przekaznikowe(
  Id_infrastruktury Integer NOT NULL,
  Wysokosc_masztu Varchar2(3 ) NOT NULL,
  Ilosc_anten Varchar2(2 ) NOT NULL,
  Zasięg Varchar2(5 ) NOT NULL,
  Liczba_klientow Varchar2(5 )
)
/

-- Add keys for table Stacje_przekaznikowe

ALTER TABLE Stacje_przekaznikowe ADD CONSTRAINT Unique_Identifier5 PRIMARY KEY (Id_infrastruktury)
/

-- Table Serwerownie

CREATE TABLE Serwerownie(
  Id_infrastruktury Integer NOT NULL,
  Pojemnosc_dyskow Varchar2(3 ) NOT NULL,
  Czy_backupowana Char(1 ) NOT NULL,
  Predkosc_polaczenia Varchar2(5 ) NOT NULL,
  Wilgotnosc Varchar2(2 ) NOT NULL,
  Temperatura Float NOT NULL
)
/

-- Add keys for table Serwerownie

ALTER TABLE Serwerownie ADD CONSTRAINT UI_Serwerownie PRIMARY KEY (Id_infrastruktury)
/

-- Table Agregatornie

CREATE TABLE Agregatornie(
  Id_infrastruktury Integer NOT NULL,
  Liczba_agregatorow Varchar2(2 ) NOT NULL,

```

```

    Maksymalna_moc Varchar2(5 ) NOT NULL,
    Czas_pracy Varchar2(2 ) NOT NULL,
    Zuzycie_paliwa Float NOT NULL
)
/

-- Add keys for table Agregatornie

ALTER TABLE Agregatornie ADD CONSTRAINT UI_Agregatornie PRIMARY KEY (Id_infrastruktury)
/

-- Table Operatorzy_sieci_komorkowej

CREATE TABLE Operatorzy_sieci_komorkowej(
    Id_operatora_sieci_komorkowej Integer NOT NULL,
    Nazwa_operatora Varchar2(100 ) NOT NULL,
    Data_zalozenia Date NOT NULL,
    Kapital Number(10,2) NOT NULL
)
/

-- Add keys for table Operatorzy_sieci_komorkowej

ALTER TABLE Operatorzy_sieci_komorkowej ADD CONSTRAINT Operator_Sieci_KomorkowejPK PRIMARY KEY (Id_oper
/

-- Table Pracownicy

CREATE TABLE Pracownicy(
    Id_pracownika Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    PESEL Char(11 ),
    Data_urodzenia Date NOT NULL,
    Data_zatrudnienia Date NOT NULL,
    Nr_telefonu Varchar2(9 ),
    Adres_email Varchar2(50 ),
    Id_operatora_sieci_komorkowej Integer NOT NULL,
    Nr_adresu Integer NOT NULL,
    Id_stanowiska Integer NOT NULL
)
/

-- Create indexes for table Pracownicy

CREATE INDEX IX_zatrudnia ON Pracownicy (Id_operatora_sieci_komorkowej)
/

CREATE INDEX IX_Relationship9 ON Pracownicy (Nr_adresu)
/

CREATE INDEX IX_Relationship13 ON Pracownicy (Id_stanowiska)
/

-- Add keys for table Pracownicy

```

```

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

-- Table Klienci

CREATE TABLE Klienci(
    Id_klienta Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Numer_telefonu Varchar2(9 ),
    Adres_email Varchar2(50 ) NOT NULL,
    Czy_newsletter Char(1 ) NOT NULL,
    Id_operatora_sieci_komorkowej Integer NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Klienci

CREATE INDEX IX_Posiada_klienta ON Klienci (Id_operatora_sieci_komorkowej)
/

CREATE INDEX IX_Relationship7 ON Klienci (Nr_adresu)
/

-- Add keys for table Klienci

ALTER TABLE Klienci ADD CONSTRAINT KlientPK PRIMARY KEY (Id_klienta)
/

-- Table Uslugi

CREATE TABLE Uslugi(
    Id_uslugi Integer NOT NULL,
    Czas_trwania Varchar2(3 ),
    Id_operatora_sieci_komorkowej Integer NOT NULL,
    Id_klienta Integer NOT NULL,
    Id_Listy_Uslug Integer NOT NULL
)
/

-- Create indexes for table Uslugi

CREATE INDEX IX_Oferuje ON Uslugi (Id_operatora_sieci_komorkowej)
/

CREATE INDEX IX_Korzysta ON Uslugi (Id_klienta)
/

CREATE INDEX IX_Uslugi_maja ON Uslugi (Id_Listy_Uslug)
/

-- Add keys for table Uslugi

ALTER TABLE Uslugi ADD CONSTRAINT UsługaPK PRIMARY KEY (Id_uslugi)
/

```

```

-- Table Nieruchomosci

CREATE TABLE Nieruchomosci(
    Id_nieruchomosci Integer NOT NULL,
    Rodzaj_nieruchomosci Varchar2(25 ) NOT NULL,
    Powierzchnia Varchar2(5 ) NOT NULL,
    Waznosc_badan_ppoz Date NOT NULL,
    Maksymalna_liczba_stanowisk Varchar2(3 ),
    Czy_monitorowane Char(1 ) NOT NULL,
    Id_operatora_sieci_komorkowej Integer NOT NULL,
    Nr_adresu Integer NOT NULL,
    Id_firmy_ochroniarskiej Integer
)
/

-- Create indexes for table Nieruchomosci

CREATE INDEX IX_Posiada_nieruchomosc ON Nieruchomosci (Id_operatora_sieci_komorkowej)
/

CREATE INDEX IX_Relationship8 ON Nieruchomosci (Nr_adresu)
/

CREATE INDEX IX_Nieruchomosc_ma_firme_ochroniarska ON Nieruchomosci (Id_firmy_ochroniarskiej)
/

-- Add keys for table Nieruchomosci

ALTER TABLE Nieruchomosci ADD CONSTRAINT NieruchomoscPK PRIMARY KEY (Id_nieruchomosci)
/

-- Table Infrastruktury

CREATE TABLE Infrastruktury(
    Id_infrastruktury Integer NOT NULL,
    Czy_krytyczna Char(1 ) NOT NULL,
    Wartosc Number(10,2) NOT NULL,
    Data_konserwacji Date NOT NULL,
    Czy_monitorowane Char(1 ) NOT NULL,
    Czy_zasilanie_awaryjne Char(1 ) NOT NULL,
    Id_operatora_sieci_komorkowej Integer NOT NULL,
    Nr_adresu Integer NOT NULL,
    Id_firmy_ochroniarskiej Integer
)
/

-- Create indexes for table Infrastruktury

CREATE INDEX IX_Dysponuje ON Infrastruktury (Id_operatora_sieci_komorkowej)
/

CREATE INDEX Infrastruktura_ma_adres ON Infrastruktury (Nr_adresu)
/

CREATE INDEX IX_Relationship3 ON Infrastruktury (Id_firmy_ochroniarskiej)

```

```

/

-- Add keys for table Infrastruktury

ALTER TABLE Infrastruktury ADD CONSTRAINT InfrastrukturaPK PRIMARY KEY (Id_infrastruktury)
/

-- Table Pracownicy_Uslugi

CREATE TABLE Pracownicy_Uslugi(
    Id_pracownika Integer NOT NULL,
    Id_uslugi Integer NOT NULL
)
/

-- Table Pracownicy_Infrastruktury

CREATE TABLE Pracownicy_Infrastruktury(
    Id_pracownika Integer NOT NULL,
    Id_infrastruktury Integer NOT NULL,
    Czy_administrator Char(1 ) NOT NULL
)
/

-- Table Pracownicy_Nieruchomosci

CREATE TABLE Pracownicy_Nieruchomosci(
    Id_pracownika Integer NOT NULL,
    Id_nieruchomosci Integer NOT NULL,
    Czy_administrator Char(1 ) NOT NULL
)
/

-- Table Listy_Uslug

CREATE TABLE Listy_Uslug(
    Id_Listy_Uslug Integer NOT NULL,
    Rodzaj_uslugi Varchar2(25 ) NOT NULL,
    Koszt Number(10,2),
    Czy_zdalna Char(1 ) NOT NULL,
    Czy_gwarancja Char(1 )
)
/

-- Add keys for table Listy_Uslug

ALTER TABLE Listy_Uslug ADD CONSTRAINT PK_Listy_Uslug PRIMARY KEY (Id_Listy_Uslug)
/

-- Table Adresy

CREATE TABLE Adresy(
    Id_adresu Integer NOT NULL,
    Miejscowosc Varchar2(40 ) NOT NULL,
    Ulica Varchar2(40 ) NOT NULL,
    Numer_budynku Varchar2(4 ),

```

```

    Numer_lokalu Varchar2(4 ),
    Id_poczty Integer NOT NULL
)
/

-- Create indexes for table Adresy

CREATE INDEX IX_Relationship4 ON Adresy (Id_poczty)
/

-- Add keys for table Adresy

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

-- Table Poczty

CREATE TABLE Poczty(
    Id_poczty Integer NOT NULL,
    Kod_poczty Char(6 ) NOT NULL,
    Miejscowosc_Poczty Varchar2(40 ) NOT NULL
)
/

-- Add keys for table Poczty

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

-- Table Wlasciciele

CREATE TABLE Wlasciciele(
    Id_wlasciciela Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Numer_telefonu Varchar2(9 ) NOT NULL,
    Id_operatora_sieci_komorkowej Integer NOT NULL,
    Id_adresu Integer NOT NULL
)
/

-- Create indexes for table Wlasciciele

CREATE INDEX IX_Relationship5 ON Wlasciciele (Id_operatora_sieci_komorkowej)
/

CREATE INDEX IX_Relationship6 ON Wlasciciele (Id_adresu)
/

-- Add keys for table Wlasciciele

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

-- Table Stanowiska

```

```

CREATE TABLE Stanowiska(
    Id_stanowiska Integer NOT NULL,
    Rodzaj_wyksztalcenia Varchar2(15 ) NOT NULL,
    Czy_mobilne Char(1 ) NOT NULL,
    Rodzaj_stanowiska Varchar2(25 ) NOT NULL
)
/

-- Add keys for table Stanowiska

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

-- Table Pakiety_uslug

CREATE TABLE Pakiety_uslug(
    Id_pakietu_uslug Integer NOT NULL,
    Rodzaj_pakietu Varchar2(40 ) NOT NULL,
    Cena Number(10,2) NOT NULL,
    Data_obowiazywania Date NOT NULL,
    Id_klienta Integer NOT NULL
)
/

-- Create indexes for table Pakiety_uslug

CREATE INDEX IX_Relationship1 ON Pakiety_uslug (Id_klienta)
/

-- Add keys for table Pakiety_uslug

ALTER TABLE Pakiety_uslug ADD CONSTRAINT PK_Pakiety_uslug PRIMARY KEY (Id_pakietu_uslug)
/

-- Table Firmy_ochroniarskie

CREATE TABLE Firmy_ochroniarskie(
    Id_firmy_ochroniarskiej Integer NOT NULL,
    Nazwa Varchar2(30 ) NOT NULL,
    Czy_oferuje_monitoring Char(1 ) NOT NULL,
    awaryjny_numer_telefonu Varchar2(9 ) NOT NULL,
    Id_adresu Integer NOT NULL
)
/

-- Create indexes for table Firmy_ochroniarskie

CREATE INDEX IX_Relationship2 ON Firmy_ochroniarskie (Id_adresu)
/

-- Add keys for table Firmy_ochroniarskie

ALTER TABLE Firmy_ochroniarskie ADD CONSTRAINT PK_Firmy_ochroniarskie PRIMARY KEY (Id_firmy_ochroniarskiej)
/

-- Trigger for sequence IdOperatoraSieciKomorkowejSEQ for column Id_operatora_sieci_komorkowej in table

```

```

CREATE OR REPLACE TRIGGER ts_Operatorzy_sieci_komorkowej_IdOperatoraSieciKomorkowejSEQ BEFORE INSERT
ON Operatorzy_sieci_komorkowej FOR EACH ROW
BEGIN
    :new.Id_operatora_sieci_komorkowej := IdOperatoraSieciKomorkowejSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Operatorzy_sieci_komorkowej_IdOperatoraSieciKomorkowejSEQ AFTER UPDATE OF
ON Operatorzy_sieci_komorkowej FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_operatora_sieci_komorkowej in table Operatorzy_sieci_komorkowej');
END;
/

-- Trigger for sequence IdPracownikaSEQ for column Id_pracownika in table Pracownicy -----
CREATE OR REPLACE TRIGGER ts_Pracownicy_IdPracownikaSEQ BEFORE INSERT
ON Pracownicy FOR EACH ROW
BEGIN
    :new.Id_pracownika := IdPracownikaSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Pracownicy_IdPracownikaSEQ AFTER UPDATE OF Id_pracownika
ON Pracownicy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_pracownika in table Pracownicy as it uses sequence');
END;
/

-- Trigger for sequence IdKlientaSEQ for column Id_klienta in table Klienci -----
CREATE OR REPLACE TRIGGER ts_Klienci_IdKlientaSEQ BEFORE INSERT
ON Klienci FOR EACH ROW
BEGIN
    :new.Id_klienta := IdKlientaSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Klienci_IdKlientaSEQ AFTER UPDATE OF Id_klienta
ON Klienci FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_klienta in table Klienci as it uses sequence');
END;
/

-- Trigger for sequence IdUslugiSEQ for column Id_uslugi in table Uslugi -----
CREATE OR REPLACE TRIGGER ts_Uslugi_IdUslugiSEQ BEFORE INSERT
ON Uslugi FOR EACH ROW
BEGIN
    :new.Id_uslugi := IdUslugiSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Uslugi_IdUslugiSEQ AFTER UPDATE OF Id_uslugi
ON Uslugi FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_uslugi in table Uslugi as it uses sequence.')
```

```

CREATE OR REPLACE TRIGGER ts_Nieruchomosci_IdNieruchomosciSEQ BEFORE INSERT
ON Nieruchomosci FOR EACH ROW
BEGIN
    :new.Id_nieruchomosci := IdNieruchomosciSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Nieruchomosci_IdNieruchomosciSEQ AFTER UPDATE OF Id_nieruchomosci
ON Nieruchomosci FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_nieruchomosci in table Nieruchomosci as it us
END;
/

-- Trigger for sequence IdInfrastrukturySEQ for column Id_infrastruktury in table Infrastruktury -----
CREATE OR REPLACE TRIGGER ts_Infrastruktury_IdInfrastrukturySEQ BEFORE INSERT
ON Infrastruktury FOR EACH ROW
BEGIN
    :new.Id_infrastruktury := IdInfrastrukturySEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Infrastruktury_IdInfrastrukturySEQ AFTER UPDATE OF Id_infrastruktury
ON Infrastruktury FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_infrastruktury in table Infrastruktury as it
END;
/

-- Trigger for sequence IdLlistyUslugSEQ for column Id_Listy_Uslug in table Listy_Uslug -----
CREATE OR REPLACE TRIGGER ts_Listy_Uslug_IdLlistyUslugSEQ BEFORE INSERT
ON Listy_Uslug FOR EACH ROW
BEGIN
    :new.Id_Listy_Uslug := IdLlistyUslugSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Listy_Uslug_IdLlistyUslugSEQ AFTER UPDATE OF Id_Listy_Uslug
ON Listy_Uslug FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_Listy_Uslug in table Listy_Uslug as it uses s
END;
/

-- Trigger for sequence IdAdresuSEQ for column Id_adresu in table Adresy -----
CREATE OR REPLACE TRIGGER ts_Adresy_IdAdresuSEQ BEFORE INSERT
ON Adresy FOR EACH ROW
BEGIN
    :new.Id_adresu := IdAdresuSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Adresy_IdAdresuSEQ AFTER UPDATE OF Id_adresu
ON Adresy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_adresu in table Adresy as it uses sequence.')
```

```

END;
/

-- Trigger for sequence IdPocztySEQ for column Id_poczty in table Poczty -----

```

```

CREATE OR REPLACE TRIGGER ts_Poczty_IdPocztySEQ BEFORE INSERT
ON Poczty FOR EACH ROW
BEGIN
    :new.Id_poczty := IdPocztySEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Poczty_IdPocztySEQ AFTER UPDATE OF Id_poczty
ON Poczty FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_poczty in table Poczty as it uses sequence.')
```

```

END;
/

-- Trigger for sequence IdWlascicielaSEQ for column Id_wlasciciela in table Wlasciciele -----
CREATE OR REPLACE TRIGGER ts_Wlasciciele_IdWlascicielaSEQ BEFORE INSERT
ON Wlasciciele FOR EACH ROW
BEGIN
    :new.Id_wlasciciela := IdWlascicielaSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wlasciciele_IdWlascicielaSEQ AFTER UPDATE OF Id_wlasciciela
ON Wlasciciele FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_wlasciciela in table Wlasciciele as it uses s
```

```

END;
/

-- Trigger for sequence IdStanowiskaSEQ for column Id_stanowiska in table Stanowiska -----
CREATE OR REPLACE TRIGGER ts_Stanowiska_IdStanowiskaSEQ BEFORE INSERT
ON Stanowiska FOR EACH ROW
BEGIN
    :new.Id_stanowiska := IdStanowiskaSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Stanowiska_IdStanowiskaSEQ AFTER UPDATE OF Id_stanowiska
ON Stanowiska FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_stanowiska in table Stanowiska as it uses seq
```

```

END;
/

-- Trigger for sequence IdPakietuUslugSEQ for column Id_pakietu_uslug in table Pakiety_uslug -----
CREATE OR REPLACE TRIGGER ts_Pakiety_uslug_IdPakietuUslugSEQ BEFORE INSERT
ON Pakiety_uslug FOR EACH ROW
BEGIN
    :new.Id_pakietu_uslug := IdPakietuUslugSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Pakiety_uslug_IdPakietuUslugSEQ AFTER UPDATE OF Id_pakietu_uslug
ON Pakiety_uslug FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_pakietu_uslug in table Pakiety_uslug as it us
```

```

END;
/

-- Trigger for sequence IdFirmyOchroniarskiejSEQ for column Id_firmy_ochroniarskiej in table Firmy_ochr
```

```

CREATE OR REPLACE TRIGGER ts_Firmy_ochroniarskie_IdFirmyOchroniarskiejSEQ BEFORE INSERT
ON Firmy_ochroniarskie FOR EACH ROW
BEGIN
    :new.Id_firmy_ochroniarskiej := IdFirmyOchroniarskiejSEQ.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Firmy_ochroniarskie_IdFirmyOchroniarskiejSEQ AFTER UPDATE OF Id_firmy_och
ON Firmy_ochroniarskie FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_firmy_ochroniarskiej in table Firmy_ochroniar
END;
/

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

ALTER TABLE Pracownicy ADD CONSTRAINT Operatorzy_zatrudniaja_pracownikow FOREIGN KEY (Id_operatora_siec
/

ALTER TABLE Klienci ADD CONSTRAINT Operatorzy_posiadaja_klientow FOREIGN KEY (Id_operatora_sieci_komork
/

ALTER TABLE Uslugi ADD CONSTRAINT Operatorzy_oferuja_uslugi FOREIGN KEY (Id_operatora_sieci_komorkowej)
/

ALTER TABLE Infrastruktury ADD CONSTRAINT Operator_dysponuje_infrastruktura FOREIGN KEY (Id_operatora_s
/

ALTER TABLE Nieruchomosci ADD CONSTRAINT Operatorzy_posiadaja_nieruchomosci FOREIGN KEY (Id_operatora_s
/

ALTER TABLE Uslugi ADD CONSTRAINT Klientici_korzystaja_z_uslug FOREIGN KEY (Id_klienta) REFERENCES Klient
/

ALTER TABLE Uslugi ADD CONSTRAINT Uslugi_maja_listy_uslug FOREIGN KEY (Id_Listy_Uslug) REFERENCES Listy
/

ALTER TABLE Adresy ADD CONSTRAINT Adresy_maja_poczty FOREIGN KEY (Id_poczty) REFERENCES Poczty (Id_pocz
/

```

```
ALTER TABLE Klienci ADD CONSTRAINT Klienci_maja_adresy FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Id_adresu) /
```

```
ALTER TABLE Nieruchomosci ADD CONSTRAINT Nieruchomosci_maja_adresy FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Id_adresu) /
```

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

```
ALTER TABLE Infrastruktury ADD CONSTRAINT Infrastruktura_ma_adres FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Id_adresu) /
```

```
ALTER TABLE Stacje_przekaznikowe ADD CONSTRAINT Infrastruktura_Stacja_przekaznikowa FOREIGN KEY (Id_infrastruktury) REFERENCES Infrastruktury (Id_infrastruktury) /
```

```
ALTER TABLE Agregatornie ADD CONSTRAINT Infrastruktura_Agregatornia FOREIGN KEY (Id_infrastruktury) REFERENCES Infrastruktury (Id_infrastruktury) /
```

```
ALTER TABLE Serwerownie ADD CONSTRAINT Infrastruktura_Serwerownia FOREIGN KEY (Id_infrastruktury) REFERENCES Infrastruktury (Id_infrastruktury) /
```

```
ALTER TABLE Wlasciciele ADD CONSTRAINT Operator_ma_wlasciciela FOREIGN KEY (Id_operatora_sieci_komorkowej) REFERENCES Operatorzy (Id_operatora_sieci_komorkowej) /
```

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

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownicy_maja_stanowiska FOREIGN KEY (Id_stanowiska) REFERENCES Stanowiska (Id_stanowiska) /
```

```
ALTER TABLE Firmy_ochroniarskie ADD CONSTRAINT Firma_ochroniarska_ma_adres FOREIGN KEY (Id_adresu) REFERENCES Adresy (Id_adresu) /
```

```
ALTER TABLE Infrastruktury ADD CONSTRAINT Infrastruktura_ma_firme_ochroniarska FOREIGN KEY (Id_firmy_ochroniarskiej) REFERENCES Firmy_ochroniarskie (Id_firmy_ochroniarskiej) /
```

/

```
ALTER TABLE Nieruchomosci ADD CONSTRAINT Nieruchomosci_maja_firme_ochroniarska FOREIGN KEY (Id_firmy_oc.  
/
```

```
ALTER TABLE Pakiety_uslug ADD CONSTRAINT Klienci_posiadaja_pakiety_uslug FOREIGN KEY (Id_klienta) REFER  
/
```

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

```
INSERT INTO Operatorzy_sieci_komorkowej (nazwa_operatora, Data_zalozenia,Kapital)  
VALUES('JanWszo', TO_DATE('2022-12-09','YYYY-MM-DD'), '98765');
```

```
INSERT INTO Poczty(kod_poczty, miejscowosc_poczty)  
VALUES('00-001', 'Warszawa');
```

```
INSERT INTO Poczty(kod_poczty, miejscowosc_poczty)  
VALUES('05-800', 'Pruszkow');
```

```
INSERT INTO Poczty(kod_poczty, miejscowosc_poczty)  
VALUES('05-806', 'Komorow');
```

```
INSERT INTO Poczty(kod_poczty, miejscowosc_poczty)  
VALUES('05-816', 'Michalowice');
```

```
INSERT INTO Poczty(kod_poczty, miejscowosc_poczty)  
VALUES('05-820', 'Piastow');
```

```
INSERT INTO Poczty(kod_poczty, miejscowosc_poczty)  
VALUES('05-530', 'Gora Kalwaria');
```

```
INSERT INTO Poczty(kod_poczty, miejscowosc_poczty)  
VALUES('05-555', 'Tarczyn');
```

```
INSERT INTO Adresy(Miejscowosc, Ulica, Numer_budynku, Numer_lokalu, Id_poczty)  
VALUES('Warszawa', 'placPolitechniki', '1', '', '1');
```

```
INSERT INTO Adresy(Miejscowosc, Ulica, Numer_budynku, Numer_lokalu, Id_poczty)  
VALUES('Reguly', 'Ulica', '3', '2', '4');
```

```
INSERT INTO Adresy(Miejscowosc, Ulica, Numer_budynku, Numer_lokalu, Id_poczty)  
VALUES('Pecice', 'UlicaWPecicach', '42', '', '4');
```

```
INSERT INTO Adresy(Miejscowosc, Ulica, Numer_budynku, Numer_lokalu, Id_poczty)  
VALUES('Gora kalwaria', 'ulicaWGorzeKalwari', '16', '2', '6');
```

```
INSERT INTO Adresy(Miejscowosc, Ulica, Numer_budynku, Numer_lokalu, Id_poczty)
VALUES('Tarczyn', 'Tarczynska', '21', '2', '7');
```

```
INSERT INTO Firmy_ochroniarskie (nazwa, czy_oferuje_monitoring, awaryjny_numer_telefonu, Id_adresu)
VALUES('OchronaNazwa', '1', '123456789', '2');
```

```
INSERT INTO KLIENCI (Imie, Nazwisko, numer_telefonu, adres_email, czy_newsletter, Id_operatora_sieci_komorkowej, Id_adresu)
VALUES('Filip', 'Polski', '192837470', 'test@wp.pl', '1', '1', '3');
```

```
INSERT INTO Pakiety_uslug (Rodzaj_pakietu, Cena, Data_obowiazywania, Id_klienta )
VALUES('Nielimitowany', '40', TO_DATE('2023-03-12','YYYY-MM-DD'), '1');
```

```
INSERT INTO Wlasciciele (Imie, Nazwisko, Numer_telefonu, Id_operatora_sieci_komorkowej, Id_adresu)
VALUES('Jan', 'Kowalski', '987654321', '1', '4');
```

```
INSERT INTO Infrastruktury (Czy_krytyczna, Wartosc, Data_konserwacji, czy_monitorowane, czy_zasilanie_awaryjne, Id_adresu)
VALUES('1', '123000', TO_DATE('2022-12-31','YYYY-MM-DD'), '1', '1', '1', '5', '1');
```

```
INSERT INTO Stacje_przekaznikowe (Id_Infrastruktury, Wysokosc_masztu, Ilosc_anten, zasięg, Liczba_klientow)
VALUES('1', '40', '15', '10', '4200');
```

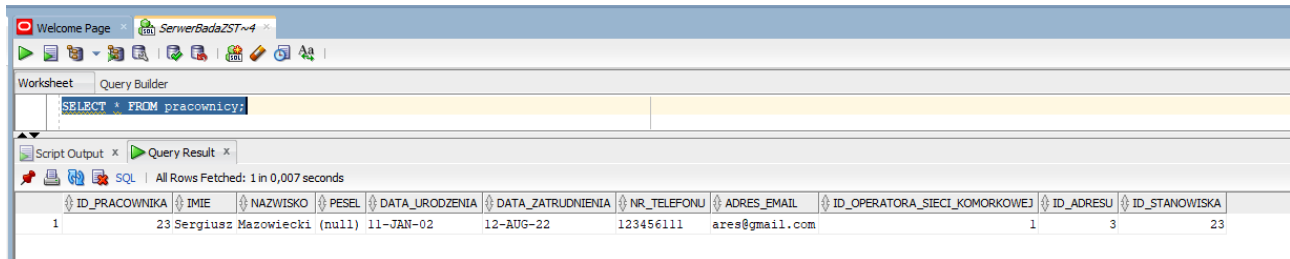
```
INSERT INTO Stanowiska (Rodzaj_wykształcenia, Czy_mobilne, Rodzaj_stanowiska)
VALUES('Wyzsze', '0', 'Sprzedawca');
```

```
INSERT INTO Pracownicy (Imie, Nazwisko, pesel, Data_urodzenia, data_zatrudnienia, nr_telefonu, adres_email)
VALUES('Sergiusz', 'Mazowiecki', '', TO_DATE('2002-01-11','YYYY-MM-DD'), TO_DATE('2022-08-12','YYYY-MM-DD'), '123456789', 'sergiusz.mazowiecki@wp.pl');
```

```
INSERT INTO Nieruchomosci (Rodzaj_nieruchomosci, powierzchnia, waznosc_badan_ppoz, maksymalna_liczba_stanowisk, Id_adresu)
VALUES('Biuro', '2000', TO_DATE('2023-01-10','YYYY-MM-DD'), '100', '1', '1', '1', '1');
```

Sprawdzenie możliwości wykonania wybranych transakcji w bazie danych:

- Wypisanie zawartości wybranej tabeli

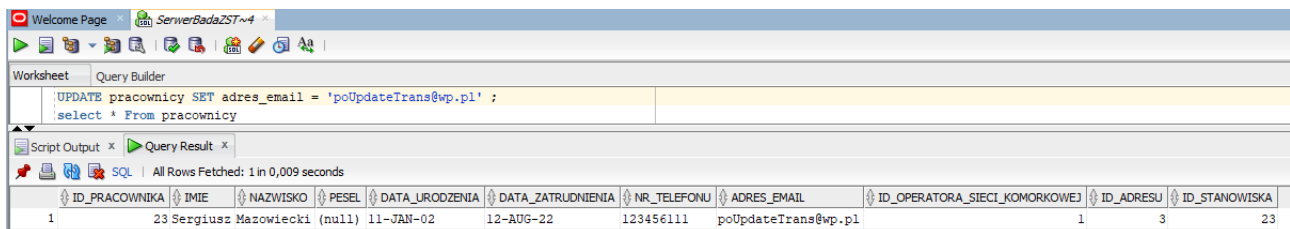


The screenshot shows the SQL Server Enterprise Manager interface. The 'Query Builder' tab is active, displaying the query: `SELECT * FROM pracownicy;`. The 'Script Output' and 'Query Result' tabs are also visible. The 'Query Result' tab shows the results of the query, which is a single row with the following data:

ID_PRACOWNIKA	IMIE	NAZWISKO	PESEL	DATA_URODZENIA	DATA_ZATRUDNIENIA	NR_TELEFONU	ADRES_EMAIL	ID_OPERATORA_SIECI_KOMORKOWEJ	ID_ADRESU	ID_STANOWISKA
1	23	Sergiusz Mazowiecki	(null)	11-JAN-02	12-AUG-22	123456111	ares@gmail.com	1	3	23

Rys. 3. Wypisanie zawartości tabeli pracownicy

- modyfikacja wartości wybranego pola

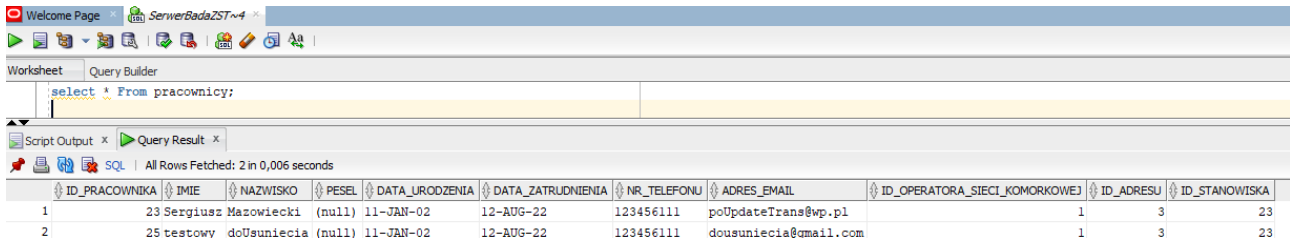


The screenshot shows the SQL Server Enterprise Manager interface. The 'Query Builder' tab is active, displaying the query: `UPDATE pracownicy SET adres_email = 'poUpdateTrans@wp.pl';`. The 'Script Output' and 'Query Result' tabs are also visible. The 'Query Result' tab shows the results of the query, which is a single row with the following data:

ID_PRACOWNIKA	IMIE	NAZWISKO	PESEL	DATA_URODZENIA	DATA_ZATRUDNIENIA	NR_TELEFONU	ADRES_EMAIL	ID_OPERATORA_SIECI_KOMORKOWEJ	ID_ADRESU	ID_STANOWISKA
1	23	Sergiusz Mazowiecki	(null)	11-JAN-02	12-AUG-22	123456111	poUpdateTrans@wp.pl	1	3	23

Rys. 4. Modyfikacja wartości pola adres_email w tabeli pracownicy

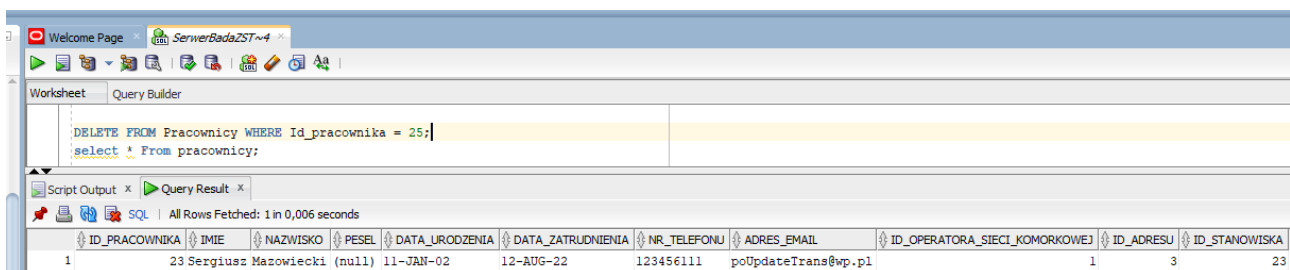
- usunięcie wybranego rekordu z tabeli



The screenshot shows the SQL Server Enterprise Manager interface. The 'Query Builder' tab is active, displaying the query: `DELETE FROM Pracownicy WHERE Id_pracownika = 25;`. The 'Script Output' and 'Query Result' tabs are also visible. The 'Query Result' tab shows the results of the query, which is a single row with the following data:

ID_PRACOWNIKA	IMIE	NAZWISKO	PESEL	DATA_URODZENIA	DATA_ZATRUDNIENIA	NR_TELEFONU	ADRES_EMAIL	ID_OPERATORA_SIECI_KOMORKOWEJ	ID_ADRESU	ID_STANOWISKA
1	23	Sergiusz Mazowiecki	(null)	11-JAN-02	12-AUG-22	123456111	poUpdateTrans@wp.pl	1	3	23
2	25	testowy doUsuniecia	(null)	11-JAN-02	12-AUG-22	123456111	dousuniecia@gmail.com	1	3	23

Rys. 5. Dodanie testowego rekordu w tabeli pracownicy



The screenshot shows the SQL Server Enterprise Manager interface. The 'Query Builder' tab is active, displaying the query: `DELETE FROM Pracownicy WHERE Id_pracownika = 25;`. The 'Script Output' and 'Query Result' tabs are also visible. The 'Query Result' tab shows the results of the query, which is a single row with the following data:

ID_PRACOWNIKA	IMIE	NAZWISKO	PESEL	DATA_URODZENIA	DATA_ZATRUDNIENIA	NR_TELEFONU	ADRES_EMAIL	ID_OPERATORA_SIECI_KOMORKOWEJ	ID_ADRESU	ID_STANOWISKA
1	23	Sergiusz Mazowiecki	(null)	11-JAN-02	12-AUG-22	123456111	poUpdateTrans@wp.pl	1	3	23

Rys. 6. Usunięcie testowego rekordu