|  |  |
| --- | --- |
| **RELACYJNE BAZY DANYCH – PROJEKT** | |
| Autor: Kamil Mogiłka | Grupa: 3 |
| Temat: Towarzystwa ubezpieczeniowe | Data: 02-06-2019 |

1. Scenariusz  
   Baza danych przeznaczona jest dla w szczególności dla podmiotów sektora ubezpieczeniowego i bankowego, jak i również dla organów nadzorujących rynek finansowy oraz podmiotów zajmujących się badaniem rynku sektora finansowego, albo branży ubezpieczeniowej/bankowej
2. Potencjalne grupy użytkowników:  
   Administrator – główny zarządca bazy danych, posiada pełen dostęp do danych  
   Organy administracyjne (np.KNF) – dostęp do wszystkich tabel baz danych w formie odczytu  
   Podmioty   
   Banki, Towarzystwa – obsługa bazy danych swoich klient, pełen dostęp
3. Wymagania funkcjonalne:

* Baza danych ma przechowywać informacje o towarzystwach ubezpieczeniowych, obsługiwanych przez nich klientach, zawartych polisach oraz informacje dotyczące doradca finansowego jako sprzedawcy oraz jego uprawnień.
* Zgromadzone dane mają umożliwić realizację zadań poprzez:  
  -kontrole oraz wsparcie pracy doradcy finansowego – ilość aktywnie obsługiwanych klientów, liczba zrealizowanych polis ubezpieczeniowych, informacje dotyczące daty wygaśnięcia polisy/zwiększonych zarobków klienta.  
  -przeprowadzania badania rynku sektora ubezpieczeń poprzez udostępnienie odpowiednich informacji o stanie lub rodzaju sprzedanych polis, ilości klientów przypadających na poszczególne towarzystwa, czy też ilość aktywnie działających sprzedawców

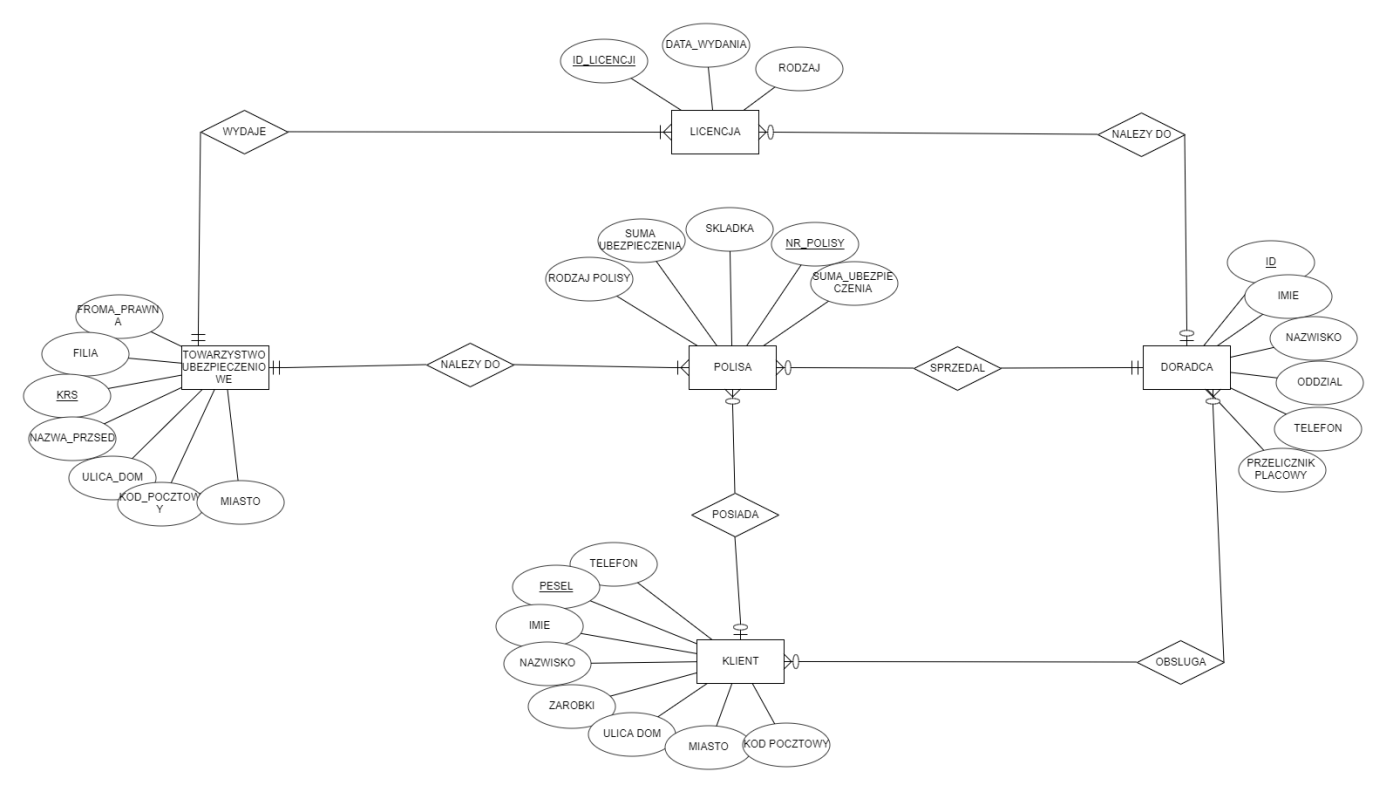
1. Wymagania niefunkcjonalne:  
   System zarządzania bazą danych – PostgreSQL  
   Kodowanie utf-8
2. 5.Diagram związków encji:  
   

Diagram składa się z 5 encji: Towarzystwo ubezpieczeniowe, licencja, doradca, polisa, klient, każda encja posiada od 3-8 atrybutów. Poszczególne encje połączone są związkami:

* Towarzystwo ubezpieczeniowe <-> Polisa - relacja jeden do wielu. Oznacza to, że jedno towarzystwo jest może, a raczej musi posiadać jedną lub wiele sprzedanych polis, natomiast polisa może należeć tylko do jednego towarzystwa.
* Polisa <-> klient – relacja jeden do wielu. Oznacza to, że klient może posiadać wiele różnych polis ubezpieczeniowych, natomiast polisa może być wykupiona tylko przez jednego klienta.
* Klient<-> Doradca – relacja wiele do wielu. Oznacza, że jeden klient może posiadać wielu doradców finansowych (każdy specjalizuje się w innych umowach), tak samo jeden doradca może mieć więcej niż jednego klienta w obsłudze.
* Relacja Doradca <-> Licencja – jeden do wielu. Jeden doradca jest w stanie posiadać wiele różnych licencji w wielu towarzystwach na przeróżne polisy, natomiast dana licencja może posiadać tylko jednego właściciela.
* Relacja Polisa <-> Doradca – jeden do wielu. Polisa może być sprzedana tylko przez jednego doradcę finansowego, natomiast doradca może sprzedać wiele polis wielu klientom.
* Relacja Licencja <-> Towarzystwo ubezpieczeniowe – jeden do wielu. Jedno towarzystwo może wydać wiele licencji dla wielu doradców, natomiast dana licencja będzie przypisana tylko do jednego towarzystwa ubezpieczeniowego

1. Przykłady zawartości najważniejszych tabel:

Przykład tabeli klient:  

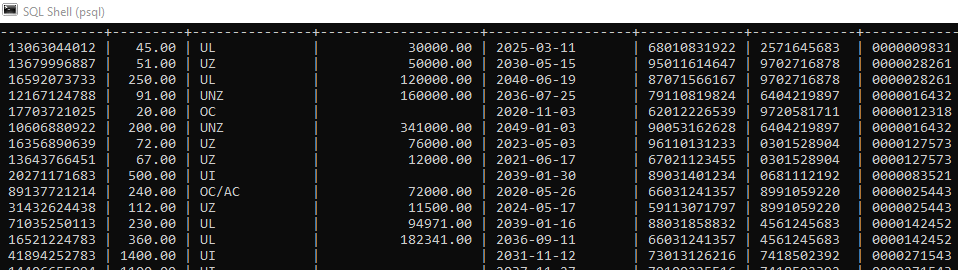

Tabela polis ubezpieczeniowych:  


Tabela doradcy finansowego:

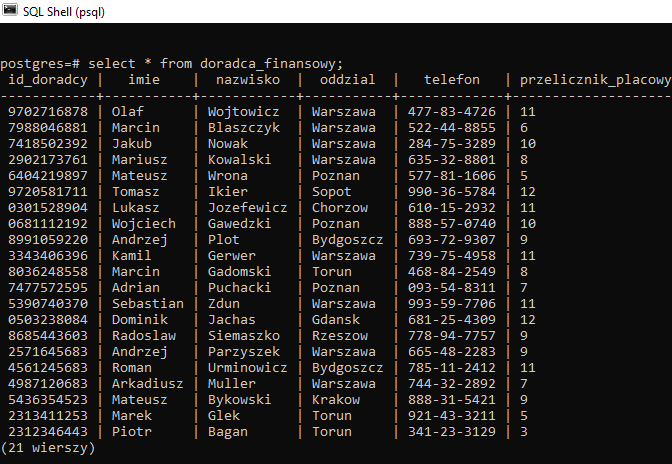
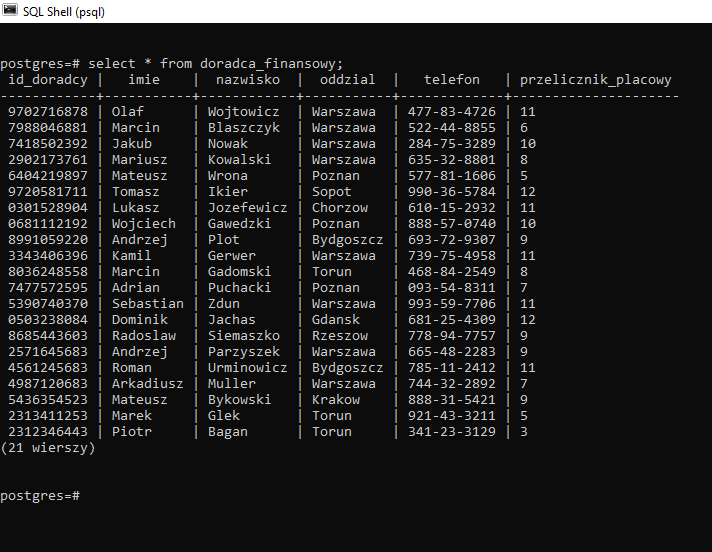


Tabela towarzystwa ubezpieczeniowego:



7.Przykłady kilku zapytań

--WYSWIETL DORADCOW I ICH KLIENTOW

SELECT DORADCA\_FINANSOWY.IMIE || ' || DORADCA\_FINANSOWY.NAZWISKO AS DORADCA, DORADCA\_FINANSOWY.ID\_DORADCY,

KLIENT.IMIE || ' ' || KLIENT.NAZWISKO AS KLIENT,

KLIENT.PESEL FROM (

(DORADCA\_FINANSOWY INNER JOIN

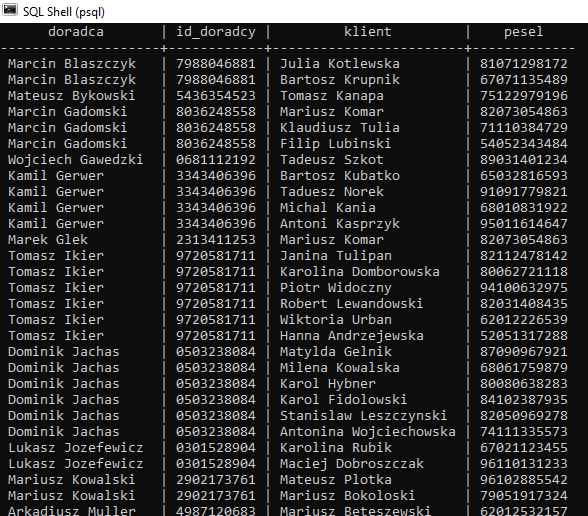
JEST\_OBSLUGIWANY

ON DORADCA\_FINANSOWY.ID\_DORADCY = JEST\_OBSLUGIWANY.ID\_DORADCY\_NR)

INNER JOIN KLIENT

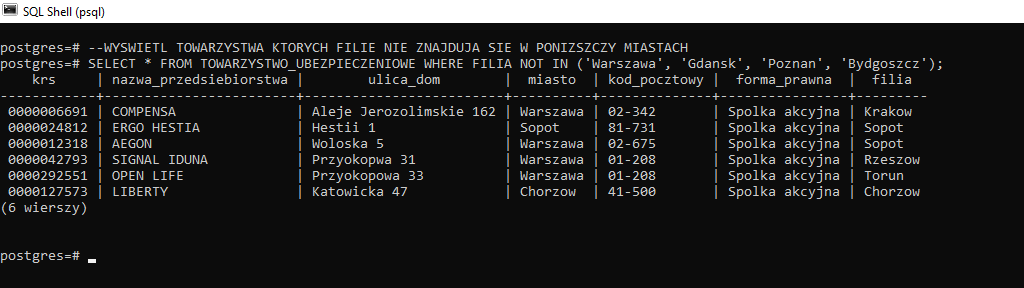
ON JEST\_OBSLUGIWANY.PESEL\_NR = KLIENT.PESEL)

ORDER BY DORADCA\_FINANSOWY.NAZWISKO;



--WYSWIETL TOWARZYSTWA KTORYCH FILIE NIE ZNAJDUJA SIE W PONIZSZCZY MIASTACH

SELECT \* FROM TOWARZYSTWO\_UBEZPIECZENIOWE WHERE FILIA NOT IN ('Warszawa', 'Gdansk', 'Poznan', 'Bydgoszcz');



--WYSWIETL DORADCE KTORY NIE POSIADA ZADNEGO KLIENTA W BAZIE

SELECT DORADCA\_FINANSOWY.IMIE || ' ' || DORADCA\_FINANSOWY.NAZWISKO AS BEZ\_KLIENTOW FROM DORADCA\_FINANSOWY WHERE ID\_DORADCY NOT IN(SELECT ID\_DORADCY\_NR FROM JEST\_OBSLUGIWANY);

