Spis treści

[ZAŁOŻENIA 2](#_Toc134395689)

[Tabele: 2](#_Toc134395690)

[Relacje: 2](#_Toc134395691)

[Schemat relacyjny 3](#_Toc134395692)

[INSTALACJA 4](#_Toc134395693)

[OPERACJE HURTOWE 4](#_Toc134395694)

[Baza imion i nazwisk 4](#_Toc134395695)

[Hurtowe dodawanie agentów: 4](#_Toc134395696)

[Hurtowe dodawanie polis: 4](#_Toc134395697)

[Hurtowe dodawanie szkód: 5](#_Toc134395698)

[GENERATORY 5](#_Toc134395699)

[OPERACJE DETALICZNE 5](#_Toc134395700)

[Dodawanie agenta: 5](#_Toc134395701)

[Dodawanie polisy: 6](#_Toc134395702)

[Dodawanie szkody: 6](#_Toc134395703)

[FUNKCJONALNOŚCI BAZY 6](#_Toc134395704)

[Generowanie widoków 6](#_Toc134395705)

[Widoki 6](#_Toc134395706)

[CIEKAWE MIEJSCA w PROGRAMIE 7](#_Toc134395707)

[Tabela Kontrahenci 7](#_Toc134395708)

[Triggery LOGON I LOGOFF 8](#_Toc134395709)

# ZAŁOŻENIA

Projekt został wykonany w środowisku Oracle Express Edition 21c

Tematem jest baza danych, która symuluje (w dużym uproszczeniu) działanie towarzystwa ubezpieczeniowego.

Możliwe jest dodawanie, usuwanie oraz modyfikacja polis, agentów oraz osób wraz z powiązanymi danymi z innych tabel.

Każda polisa jest powiązana z 1 agentem, polisa może obejmować 1 lub wiele osób.

Każdy agent może wprowadzić wiele polis.

Do każdej polisy są przypisane osoby: jedna jako ubezpieczający oraz jedna lub wiele jako ubezpieczeni.

Z każdej polisy można zgłosić jedną lub wiele szkód.

Szkoda ma 4 możliwe statusy: zgłoszona (1) , rozpatrywana (2), odrzucona (3), wypłacona (4).

Szkoda powinna być rozpatrzona w czasie 14 dni od zgłoszenia

Baza zapisuje w tabeli info\_log informacje dotyczące logowania i wylogowania użytkowników (triggery LOGON, LOGOFF)

Baza zapisuje w tabeli info\_dane informacje dotyczące pracy z danymi w tabelach agenci, polisy, szkody (trigger DML)

Baza 1 raz dziennie uruchamia job o nazwie „job\_stat” który odświeża statystyki tabel faktów (tabele polisy, kontrahenci, szkody) oraz zapisuje dane o liczbie ich wierszy do tabeli stat\_info.

Baza 1 raz dziennie odświeża widok zmaterializowany o nazwie „mv\_polisy\_koniec” który zawiera informacje nt. polis oraz ich właścicieli, dla polis których termin ważności upływa w ciągu 7 dni

## Tabele:

agenci

imiona\_meskie - baza imion

imiona\_zenskie - baza imion

info\_log - info na temat logowania / wylogowania

kontrahenci - tabela pośrednia polisy-osoby

mv\_polisy\_koniec - widok zmaterializowany – kończące się polisy

nazwiska\_meskie - baza nazwisk

nazwiska\_zenskie - baza nazwisk

osoby

polisy (podzielona na partycje względem lat)

rola

stat\_info - tabela z informacjami dot. statystyk fact tables

szkody (podzielona na partycje względem lat)

szkody\_status - opisy statusów szkód

## Relacje:

agenci polisy 1:M

osoby kontrahenci 1:M

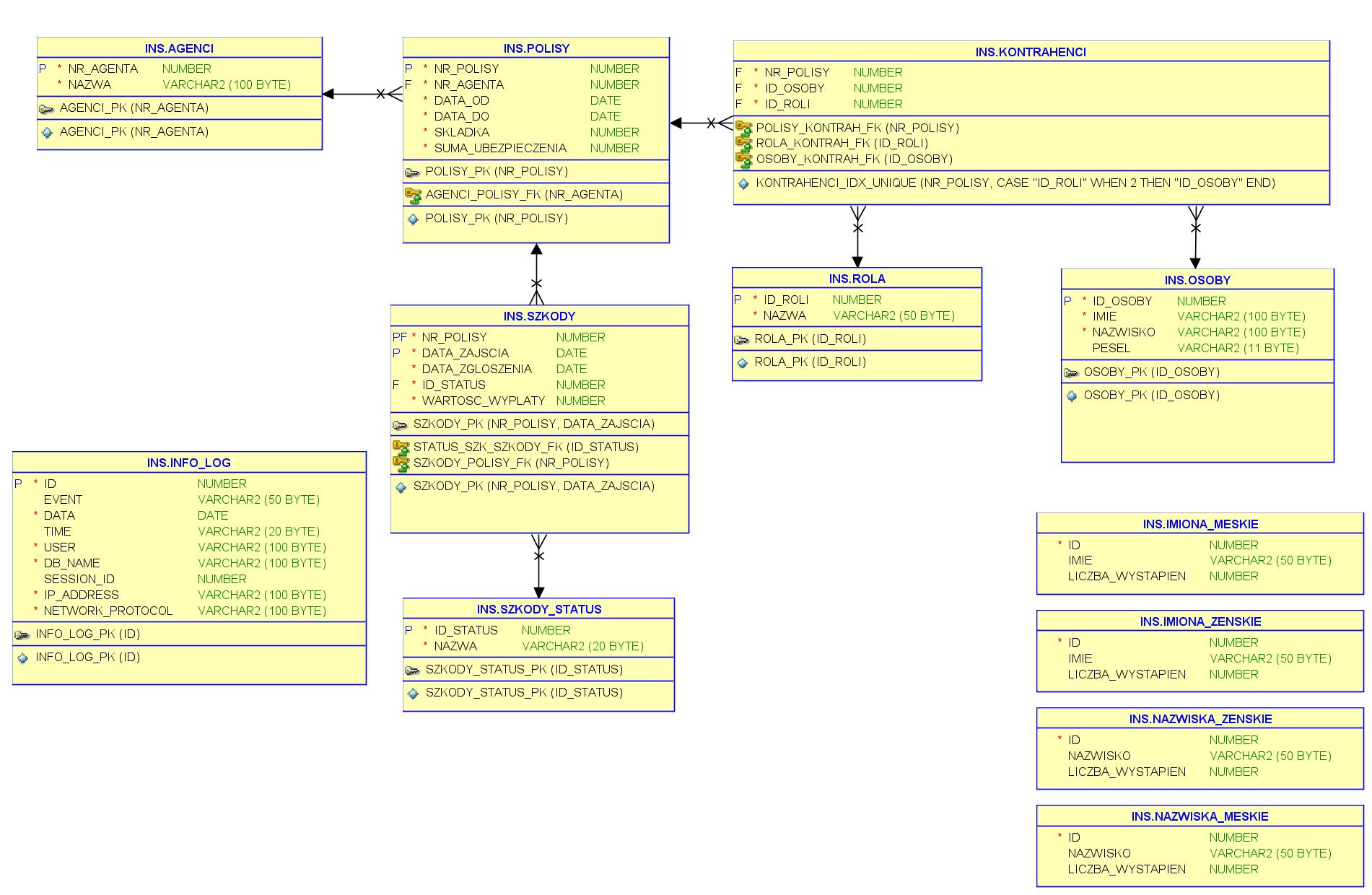
polisy kontrahenci 1:M

polisy szkody 1:M

szkody\_status szkody 1:M

Baza posiada możliwość indywidualnego oraz hurtowego (w celu szybkiego uzyskania dużej bazy danych) dodawania zdarzeń.

## Schemat relacyjny



# INSTALACJA

Głównym plikiem instalacyjnym jest plik 00\_create\_db.sql, który z kolei uruchamia pozostałe pliki.

W pliku tym w sekcji "ustawienia bazy" należy ustawić własne parametry.

Instalacja bazy danych odbywa się poprzez uruchomienie w bazie danych skryptu z pliku 00\_create.sql

Instalacja na samym początku sprawdza czy istnieje podany schemat użytkownika w wybranej lokalizacji i jeśli tak to go usuwa po czym tworzy od nowa.

Ustawienia językowe bazy danych to EE8PC852

# OPERACJE HURTOWE

## Baza imion i nazwisk

Baza korzysta z tabel imion i nazwisk które powstały poprzez przeniesienie pierwszych 2000 wierszy z zewnętrznej bazy imion oraz nazwisk (pliki csv)

https://dane.gov.pl/pl/dataset/1681,nazwiska-osob-zyjacych-wystepujace-w-rejestrze-pesel

https://dane.gov.pl/pl/dataset/1501

Dane źródłowe są posortowane od najczęściej do najrzadziej występujących.

Najpierw tworzone są tabele zewnętrzne (external tables) aby uzyskać dostęp do danych z plików \*.csv

następnie na ich podstawie tworzone są właściwe tabele ograniczone do 2000 wierszy z najczęściej występującymi imionami i nazwiskami

**Tabele**

- imiona\_meskie

- imiona\_zenskie

- nazwiska\_meskie

- nazwiska\_zenskie

## Hurtowe dodawanie agentów:

**Pakiet agenci\_pkg\_dodaj\_agentow\_hurt(p\_nazwa\_agenta:=’Agent’, p\_ilosc:=1, p\_autonum:=TRUE)**

Składa się z funkcji publicznych i prywatnych

Sprawdza max numer agenta w tabeli agenci i dodaje na kolejnych miejscach nowych agentów.

Domyślna nazwa to ‘Agent’

Domyślnie dodawana jest autonumeracja po nazwie

Domyślne nazwy agentów z autonumeracją to 'Agent 00x' , gdzie 'x' = nr\_agenta (generowany przez sekwencję identity)

**Funkcje pakietu:**

- dodaj agentow\_hurt (pub) - dodaj podaną liczbę agentów do tabeli agenci

- dodaj agenta (pub) - dodaj podaną liczbę agentów do tabeli agenci

- ustaw\_seq (priv) - ustawia seq na ostatni nr\_agenta w tabeli agenci

## Hurtowe dodawanie polis:

**Pakiet polisy\_pkg.dodaj\_polisy\_hurt (ilosc\_polis, data min, data max, ilosc\_osob) - do zrobienia**

Składka jest liczona jako 10% sumy ubezpieczenia

Ilość osób - ile max osób może być dodanych na polisach (od 0 do ilosc\_osob)

Dodanie polisy musi dodawać co najmniej 2 wpisy w tabeli kontrahenci (1 osoba która kupuje polisę jako ubezpieczający i 1 osoba jako ubezpieczony

**Pakiet osoby\_pkg.dodaj\_osoby\_hurt(ilosc osob) – do zrobienia**

Dla każdej polisy generowana jest losowa ilość osób z podanego zakresu,może być więcej ubezpieczonych.

Na tej podstawie generowane są osoby, które są dodawane do polisy jako ubezpieczony (tabela osoby i kontrahenci).

## Hurtowe dodawanie szkód:

**Pakiet szkody\_pkg.dodaj\_szkody(ilosc,max\_ilosc\_szkod) – do zrobienia**

Z tabeli polisy losowane są numery polis na których zostanie zgłoszona szkoda oraz jej status

Dla każdej wylosowanej polisy zostaną wygenerowana od 1 do n ilość szkód (n=max\_ilość\_szkód)

Tylko dla szkód ze statusem zatwierdzona zostanie wpisana wyplata w losowej kwocie z zakresu od 0 do suma\_ubezpieczenia

# GENERATORY

Baza posiada pakiet generatory\_pkg, który służy do generowania wartości potrzebnych do hurtowego wypełniania tabel. Składnikami pakietu są funkcje, które jako wynik zwracają wygenerowane wartości

Pakiet ten umożliwia generowanie następujących danych:

- generuj\_pesel: generuje dla podanego zakresu dat prawidłowy PESEL wraz z cyfrą kontrolną

Obsługuje lata 1900 – 2099.

- generuj\_dane\_osobowe: generuje dane korzystając z bazy danych imion i nazwisk, podanie płci jest

opcjonalne, ale umożliwia zawężenie wygenerowania danych do konkretnej płci, np. przy wykorzystaniu nr PESEL (10 cyfra identyfikuje płeć)

- generuj\_date: generuje datę dla podanego zakresu dat

- generuj\_sume\_ubezp: generuje sumę ubezpieczenia dla podanego zakresu

# OPERACJE DETALICZNE

## Dodawanie agenta:

**Pakiet agenci\_pkg.dodaj\_agenta(p\_nazwa\_agenta,p\_autonum:=FALSE)**

1. dodaje 1 agenta o wybranej nazwie wywołując wewnątrz procedurę dodaj\_agentow\_hurt
2. Domyślnie nie dodaje autonumeracji po nazwie

## Dodawanie polisy:

**Pakiet polisy\_pkg.dodaj\_polise**

**Pakiet osoby\_pkg.dodaj\_osobe (obsługuje wpisy do 2 tabel- osoby + kontrahenci)**

1) dodać polisę

2) dodać do polisy osoby wraz z ich rolą na polisie (tabela osoba + kontrahenci)

## Dodawanie szkody:

**Pakiet szkody\_pkg.dodaj\_szkode**

1) dodać szkodę do wybranej polisy

# FUNKCJONALNOŚCI BAZY

## Widoki zmaterializowane

1. **mv\_polisy\_koniec** – polisy których termin ważności kończy się do 7 dni Widok jest odświeżany codziennie o północy
2. mv\_kwoty\_odszkodowan – do zrobienia
   1. kwoty wypłaconych odszkodowań w podziale na m-ce wg daty zgłoszenia w wybranym zakresie czasu
3. mv\_sr\_czas\_wyplaty - – do zrobienia
   1. średni czas (w dniach) jaki upłynął od zajścia szkody do jej zgłoszenia w wybranym kwartale które zakończyły się wypłatą
4. mv\_bilans\_roczny - – do zrobienia
   1. suma wpłaconych składek do sumy wypłaconych odszkodowań w podziale na m-ce w wybranym roku

## Widoki

1. **v\_polisy\_osoby**
   1. dane wszystkich osób + liczba polis, gdzie osoby występują jako 'ubezpieczający'
2. **v\_polisy\_wlasciciele**
   1. dane tylko tych osób które są właścicielami polis + ilość powiązanych polis
3. **v\_polisy\_indywidualne**
   1. numer polis na których jest tylko 1 osoba ubezpieczona (dowolna osoba)
4. **v\_polisy\_indywiduane\_wlascicieli**
   1. numery polis dla których ubezpieczający i ubezpieczany to ta sama osoba i nie ma innych osób na polisie 🡪1 osoba na polisie i jest to ubezpieczający
5. **v\_agenci\_wskazniki**
   1. wszyscy agenci (również ci bez polis) wraz z ilością sprzedanych polis oraz wskaźnikiem, ile tych polis jest aktywnych na dzisiaj
6. **v\_szkody\_przeterminowane**
   1. szkody, które zaszły w okresie ubezpieczenia a zostały zgłoszone po zakończeniu okresu ubezpieczenia
7. **v\_szkody\_wysokie**
   1. liczba szkód oraz wartość wypłaconych odszkodowań pogrupowanych według peseli osób, które zawarły ubezpieczenia (występują w roli ubezpieczający) Na zestawieniu są tylko te pozycje, dla których łączna wartość wypłat przekracza 100 tys.
8. **v\_polisy\_przekroczona\_wartosc**
   1. polisy, do których zostały zgłoszone szkody i dla których wartość wypłaty przekracza sumę ubezpieczenia.
9. **v\_polisy\_bez\_szkod**
   1. polisy, do których nie została zgłoszona szkoda.
10. **v\_szkody\_opoznione**
    1. szkody, które zostały zgłoszone a nie zostały rozpatrzone w czasie 7 dni
11. **v\_wskazniki\_szkodowosci**
    1. stosunek (wyrażony w %) liczby polis ze szkodami do łącznej liczby polis oraz stosunek (wyrażony w %) łącznej kwoty wypłaty odszkodowania do łącznej sumy ubezpieczenia z polis, do których została zgłoszona co najmniej jedna szkoda.
12. **v\_bilans\_roczny – do zrobienia**
    1. ilość wpłaconych składek do ilości wypłaconych odszkodowań w podziale na lata

# CIEKAWE MIEJSCA w PROGRAMIE

## Tabela Kontrahenci

Jest to tabela pośrednia łącząca polisy z osobami

Pola tabeli: nr\_polisy, id\_osoby, id\_roli (1-ubezpieczający, 2-ubezpieczony).

Początkowo na tych wszystkich polach był zdefiniowany Primary Key (Composite PK)

Wymaga on jednak podania już w czasie definicji tabeli wszystkich kolumn, w których wartości mają być sprawdzane pod kątem unikalności.

Aby móc zachować założoną funkcjonalność tabeli, czyli dla każdej polisy:

- może być tylko 1 osoba jako ubezpieczający

- może być wiele osób jako ubezpieczeni

nie można użyć PK, gdyż:

dla PK(nr\_polisy)

- nie można dodać do polisy więcej niż 1 osoby

dla PK(nr\_polisy, id\_osoby)

- nie można dodać do polisy tej samej osoby jako ubezpieczający i ubezpieczony

- można dodać do polisy różne osoby jako ubezpieczający

dla PK(nr\_polisy, id\_osoby, id\_roli)

- można dodać do polisy różne osoby jako ubezpieczający

**Utworzenie Indexu Unique zamiast PK rozwiązuje sprawę:**

CREATE UNIQUE INDEX Idx\_Kontrahenci\_Unique on kontrahenci

(

nr\_polisy,

CASE WHEN id\_roli=2 THEN id\_osoby END

);

nr\_polisy jest zawsze sprawdzany

id\_osoby jest dodawane do sprawdzania gdy id\_roli=2 (ubezpieczony)

## Triggery LOGON I LOGOFF

Próba zmiany nazwy schematu na dobierany dynamicznie w zależności od nazwy użytkownika INS.SCHEMA 🡪 &&v\_user.SCHEMA spowodowała wystąpienie błędu, gdyż wyrażenie „&&v\_user.” jest interpretowane jako próba złączenia zmiennej &&v\_user ze stringiem SCHEMA. Jeżeli &&v\_user= INS to wynikiem jest INSSCHEMA zamiast INS.SCHEMA.

Poprawny zapis wygląda następująco: INS.SCHEMA 🡪 &&v\_user..SCHEMA

Pierwsza kropka to znak łączenia, druga kropka odpowiada za .SCHEMA, czyli mamy INS.SCHEMA