**Wojskowa Akademia Techniczna**

**Sprawozdanie z projektu zaliczeniowego**

**Temat:**

**Baza danych pojazdów**

Przedmiot: Bazy Danych

Prowadzący: płk dr inż. Jarosław Koszela

Wykonała: Agnieszka Kampa

Grupa: WCY21KY1S1

Nr albumu: 80452

Spis treści

[1. Treść polecenia 4](#_Toc127894893)

[2. Temat projektu i realizacja 5](#_Toc127894894)

[3. Opis modelu konceptualnego relacyjnej bazy danych 6](#_Toc127894895)

[4. Model konceptualny relacyjnej bazy danych 7](#_Toc127894896)

[4.1. Diagram modelu konceptualnego relacyjnej bazy danych 7](#_Toc127894897)

[4.2. Encje, opisy i atrybuty 8](#_Toc127894898)

[5. Opis modelu fizycznego relacyjnej bazy danych 10](#_Toc127894899)

[6. Model fizyczny relacyjnej bazy danych 11](#_Toc127894900)

[6.1 Diagram modelu fizycznego relacyjnej bazy danych 11](#_Toc127894901)

[7. Widoki 12](#_Toc127894902)

[7.1. UrzednikW 12](#_Toc127894903)

[7.2 InfoOPojezdzieW 13](#_Toc127894904)

[7.3 KtoIleWydalW 14](#_Toc127894905)

[7.4 FabrykaW 15](#_Toc127894906)

[8. Triggery 16](#_Toc127894907)

[8.1 WprowadzeniePojazdu 16](#_Toc127894908)

[8.2 WadliwaMarka 17](#_Toc127894909)

[8.3 CenaPojazdu 18](#_Toc127894910)

[8.4 DuzaIloscNapraw 19](#_Toc127894911)

[9. Procedury 20](#_Toc127894912)

[9.1. DodajOsobeProcedura 20](#_Toc127894913)

[9.2. WyswietlPojazdyProcedura 21](#_Toc127894914)

[9.3. PrzegladProcedura 22](#_Toc127894915)

[9.4. SprawdzaniePeseluProcedura 23](#_Toc127894916)

[9.5. SprawdzaniePlciProcedura 25](#_Toc127894917)

[10. Funkcje 26](#_Toc127894918)

[10.1. WojewodztwoFunkcja 26](#_Toc127894919)

[10.2. WJakimMiescieProdukcjaFunkcja 28](#_Toc127894920)

[10.3. WaznoscRejestracjiFunkcja 30](#_Toc127894921)

[10.4. WadliwaMarkaFunkcja 31](#_Toc127894922)

[10.5. UsunFabrykeFunkcja 32](#_Toc127894923)

[11. Użytkownicy 34](#_Toc127894924)

[11.1 WlascicielPojazdu 34](#_Toc127894925)

[11.2. Urzednik 34](#_Toc127894926)

[11.3. DealerSamochodow 34](#_Toc127894927)

[11.4. Zestawienie uprawnień 34](#_Toc127894928)

# 1. Treść polecenia

Utworzyć bazę danych na dowolnie wybrany temat. Baza powinna składać się z:

* 1. minimum 10 tabel
  2. minimum 3 perspektyw (w tym co najmniej jedna zmaterializowana i funkcyjna)
  3. minimum 3 triggerów/procedur wyzwolonych
  4. minimum 3 funkcji (w tym co najmniej jedna z wykorzystaniem kursora oraz co najmniej jedna z wykorzystaniem transakcji)
  5. minimum 3 procedur (w tym co najmniej jedna z wykorzystaniem kursora oraz co najmniej jedna z wykorzystaniem transakcji)
  6. 3 użytkowników o różnych uprawnieniach
  7. Mechanizmu autoinkrementacji

Baza danych powinna zostać wykonana w dowolnym zaawansowanym oprogramowaniu oraz w oprogramowaniu opensource’owym

Bazę danych należało wykonać na podstawie utworzonego wcześniej modelu konceptualnego oraz fizycznego.

# 2. Temat projektu i realizacja

Tematem bazy danych jest Rejestracja Pojazdów. W bazie będą umieszczone informacje na temat osób, aktualnych właścicieli pojazdów zarejestrowanych. Są tam również dokładne dane nt. ww. pojazdów. Są to informacje o miejscu produkcji tych pojazdów, o naprawach, dodatkowym wyposażeniu oraz informacje na temat urzędów w jakich dokonano rejestracji.

W bazie danych realizowane są mechanizmy umożliwiające dodawanie urzędników oraz właścicieli, usuwanie nierentownych fabryk, obliczanie ilości wydanych pieniędzy, sprawdzanie miejsca rejestracji, kontrolowanie wprowadzanych danych.

# 3. Opis modelu konceptualnego relacyjnej bazy danych

Model konceptualny będzie używany do określenia struktury danych w bazie danych i jest niezbędny do tworzenia modelu fizycznego bazy danych.

Model konceptualny bazy danych składa się z elementów(encji), konkretnych informacji o elementach. Relacje między elementami określają zależności między nimi.

W modelu konceptualnym bazy danych o rejestracji pojazdów encje zawierają właścicieli pojazdów, pojazdy, urzędników rejestrujących pojazdy, fabryki, miejsca rejestracji pojazdów i naprawy. Każda encja będzie miała swoje atrybuty, takie jak: imię i nazwisko właściciela pojazdu, marka i model pojazdu, adres producenta pojazdu, adres urzędu rejestrującego pojazd, informacje o naprawie pojazdu itp.

Relacje między encjami będą odzwierciedlały zależności między nimi. Relacje między właścicielem a pojazdem będą wskazywać, że właściciel posiada określony pojazd. Relacje między pojazdem a producentem wskazują, że pojazd został wyprodukowany przez określonego producenta. Relacja między pojazdem a urzędnikiem rejestrującym wskazuje, że pojazd został zarejestrowany przez określonego urzędnika. Relacja między pojazdem a miejscem naprawy wskazuje, że pojazd został naprawiony w określonym miejscu. Relacja między pojazdem a miejscem rejestracji wskazuje, że pojazd został zarejestrowany w określonym miejscu.

Model konceptualny bazy danych o rejestracji pojazdów jest konieczny do stworzenia modelu fizycznego bazy danych. Model ten będzie zawierał elementy i mechanizmy umożliwiające obsługę bazy danych.

# 4. Model konceptualny relacyjnej bazy danych

4.1. Diagram modelu konceptualnego relacyjnej bazy danych



## 4.2. Encje, opisy i atrybuty

|  |  |  |
| --- | --- | --- |
| Encja | Atrybuty | Komentarz |
| Osoba | PESEL  NrPrawaJazdy  NrDowodu  DataUrodzenia  Plec | Encja zawierająca informacje o osobach w bazie które mogą mieć pojazdy. |
| Rejestracja | IDRejestracji  DataRejestracji  WaznoscRejestracji | Encja zawierająca informacje o konkretnej rejestracji samochodu w określonym momencie w urzędzie. |
| Pojazd | IDPojazdu  VIN  NrRejestracyjny  RokProdukcji  PojemnoscSilnika  Cena | Encja zawierająca informacje o konkretnym rejestrowanym pojeździe. |
| Urzednik | IDUrzednika | Encja zwierająca informacje o urzędniku przeprowadzającym rejestracje. |
| Urzad | IDUrzedu | Encja zawierająca informacje o urzędzie w którym odbywa się rejestracja. |
| Marka | IDMarki  NazwaMarki | Encja zawierająca informacje o marce samochodu. |
| DaneOsobowe | Imie  Nazwisko | Encja abstrakcyjna zawierająca dane osobowe. Dziedziczy z niej encja Naprawy, Urzędnik oraz Osoba. |
| Adres | KodPocztowy  Poczta  Miejscowosc  Ulica  NumerDomu  NumerLokalu | Encja z której dziedziczy Osoba, Urząd oraz Fabryka. Jest to encja abstrakcyjna. |
| Naprawy | IDNaprawy  CzescSamochodu  DataNaprawy  IDPracownika  KwotaNaprawy | Encja zawierająca informacje o naprawach przeprowadzanych na samochodzie. |
| Fabryka | IDFabryki | Encja zawierająca informacje o fabryce w której został wyprodukowany samochód. |
| RodzajWyposazenia | IDWyposazenia  TypWyposazenia | Encja zawierająca informację o dodatkowym wyposażeniu samochodów. |

# 5. Opis modelu fizycznego relacyjnej bazy danych

Model fizyczny bazy danych będzie zawierał informacje o tym, jakie tabele będą używane, jakie będą ich pola, jakie będą typy danych, jakie będą klucze główne, jakie będą indeksy itp.

Model fizyczny bazy danych dotyczącej rejestracji pojazdów opiera się na konceptualnym modelu bazy danych, który zawiera informacje o osobach, aktualnych i poprzednich właścicielach pojazdów, dokładnych danych na temat pojazdów, miejscu produkcji, naprawach, dodatkowym wyposażeniu i miejscach rejestracji. Model fizyczny zawiera wszystkie te informacje, ale w postaci struktury danych gotowej do użycia. Model fizyczny zawiera też widoki oraz procedury wyzwolone. W modelu fizycznym widzimy klucze obce.

W modelu fizycznym generuje się tabela DodatkoweWyposazenie, intersekcja, wynikająca z relacji wiele do wielu w modelu konceptualnym. W modelu fizycznym zostało zrealizowane dziedziczenie, a encja DaneOsobowe oraz Adres nie zostały wygenerowane.

Tak więc, głównym celem modelu fizycznego jest umożliwienie łatwego i szybkiego dostępu do wszystkich tych informacji w jednym miejscu. Ponadto, model fizyczny zapewnia także zabezpieczenia dla przechowywanych informacji, takie jak ograniczenia dostępu, ograniczenia zapytań i ograniczenia edycji.

# 6. Model fizyczny relacyjnej bazy danych

## 6.1 Diagram modelu fizycznego relacyjnej bazy danych



# 7. Widoki

W zadaniu należało zaimplementować minimum 3 widoki w tym jeden zmaterializowany

## 7.1. UrzednikW

UrzednikW to widok który wyświetla ze strony urzędnika osoby, ich zarejestrowane pojazdy oraz informację o tym ile lat dany pojazd jest już zarejestrowany.

Kod widoku:

|  |
| --- |
| ALTER VIEW "DBA"."UrzednikW"  AS  SELECT Osoba.Imie, Osoba.Nazwisko,  Rejestracja.NrRejestracyjny, Rejestracja.VIN, datediff(year, Rejestracja.DataRejestracji,  GETDATE()) AS "IleLatZarejestrowany"  FROM Osoba  INNER JOIN Rejestracja ON Rejestracja.PESEL = Osoba.PESEL  ORDER BY 1 |

Wywołanie:



Wynik wywołania:

Obraz zawierający stół

Opis wygenerowany automatycznie

## 7.2 InfoOPojezdzieW

InfoOPojezdzieW to widok, który wyświetla informację o tym kto i za ile kupił samochód oraz w jakiej fabryce został wyprodukowany.

Kod widoku:

|  |
| --- |
| ALTER VIEW "DBA"."InfoOPojezdzieW"  AS  SELECT Osoba.Nazwisko+' '+Osoba.Imie AS Wlasciciel, Pojazd.Cena,Marka.NazwaMarki, Fabryka.IDFabryki  FROM Osoba  INNER JOIN Rejestracja ON Rejestracja.PESEL=Osoba.PESEL  INNER JOIN Pojazd ON Pojazd.VIN=Rejestracja.VIN  INNER JOIN Marka ON Marka.IDMarki=Pojazd.IDMarki  INNER JOIN Fabryka ON Fabryka.IDFabryki=Pojazd.IDFabryki  ORDER BY Rejestracja.PESEL ASC |

Wywołanie:



Wynik wywołania:

Obraz zawierający tekst, stół

Opis wygenerowany automatycznie

## 7.3 KtoIleWydalW

KtoIleWydalW to widok, który wyświetla informację o tym kto ile wydał na samochód, łącznie z naprawami. Widok wyświetla każdego zarejestrowanego właściciela, nawet jeśli nie posiada on samochodów.

Kod widoku:

|  |
| --- |
| ALTER VIEW "DBA"."KtoIleWydalW"  AS  SELECT Osoba.PESEL, MAX(Osoba.Nazwisko), MAX(Osoba.Imie), COALESCE(MAX(Pojazd.Cena),0) AS "Cena Pojazdu",  SUM(COALESCE(Naprawy.KwotaNaprawy,0)) AS "Laczna Kwota Napraw"  FROM Osoba  LEFT JOIN Rejestracja ON Rejestracja.PESEL = Osoba.PESEL  LEFT JOIN Pojazd ON Pojazd.VIN = Rejestracja.VIN  LEFT JOIN Naprawy ON Naprawy.VIN=Pojazd.VIN  GROUP BY Osoba.PESEL,Pojazd.VIN |

Wywołanie:



Rezultaty:

Obraz zawierający stół

Opis wygenerowany automatycznie

## 7.4 FabrykaW

FabrykaW to zmaterializowany widok wyświetlający ile pojazdów wyprodukowała jaka fabryka.

Kod widoku:

|  |
| --- |
| CREATE MATERIALIZED VIEW "DBA"."FabrykaW"  AS  SELECT Fabryka.IDFabryki, COUNT(Pojazd.IDFabryki) AS "LiczbaWyprodukowanych",  Fabryka.Miejscowosc, COUNT(DodatkoweWyposazenie.IDWyposazenia) AS "LiczbaDodatkowego"  FROM Fabryka  LEFT JOIN Pojazd ON Pojazd.IDFabryki = Fabryka.IDFabryki  LEFT JOIN DodatkoweWyposazenie ON DodatkoweWyposazenie.VIN=Pojazd.VIN  GROUP BY Fabryka.IDFabryki,Fabryka.Miejscowosc  ORDER BY Fabryka.IDFabryki |

Wywołanie widoku:



Rezultaty:

Obraz zawierający stół

Opis wygenerowany automatycznie

# 8. Triggery

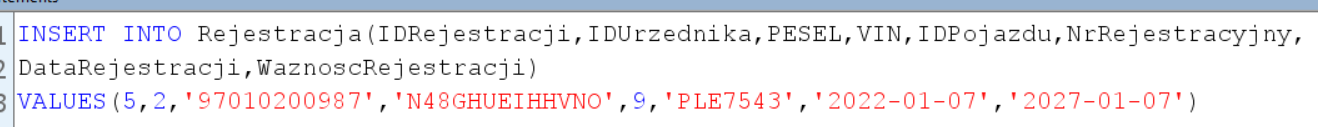
## 8.1 WprowadzeniePojazdu

Ten trigger zapobiega zarejestrowaniu pojazdu który jest wyprodukowany wczesniej niz 30 lat temu.

Kod:

|  |
| --- |
| ALTER TRIGGER "WprowadzeniePojazdu" BEFORE INSERT  ORDER 1 ON "DBA"."Rejestracja"  REFERENCING NEW AS Rejestracja  FOR EACH ROW  BEGIN  DECLARE rok DATE;    SET rok=(SELECT MAX(Pojazd.RokProdukcji) FROM Pojazd  INNER JOIN Rejestracja ON Pojazd.VIN=Rejestracja.VIN);    IF datediff(year, rok, GETDATE()) > 30  THEN RAISERROR 17001 'pojazd za stary do rejestracji';  ENDIF;  END |

Wywołanie:



Wynik:

Obraz zawierający tekst

Opis wygenerowany automatycznie

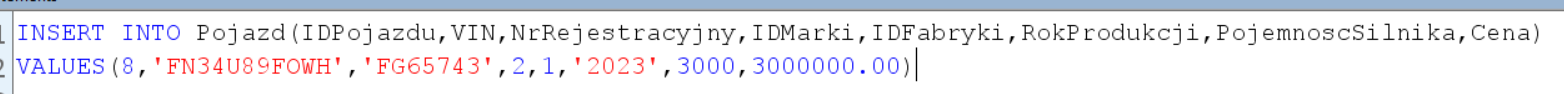
## 8.2 WadliwaMarka

Ten trigger ostrzega po wstawieniu pojazdu ktory jest najbardziej wadliwej marki.

Kod:

|  |
| --- |
| ALTER TRIGGER "WadliwaMarka" AFTER INSERT  ORDER 1 ON "DBA"."Pojazd"  REFERENCING NEW AS Pojazdd  FOR EACH ROW  BEGIN  IF Pojazdd.IDMarki=WadliwaMarkaFunkcja() THEN  MESSAGE 'uwaga! najbardziej wadliwa marka!' TO CLIENT;  ENDIF  END |

Wywołanie:



Wynik:

Obraz zawierający tekst

Opis wygenerowany automatycznie

## 8.3 CenaPojazdu

Ten trigger nie pozwala dodac pojazdu ktorego cena wykracza po za 20000 a jest starszy niz 20 lat

Kod:

|  |
| --- |
| ALTER TRIGGER "CenaPojazdu" BEFORE INSERT  ORDER 1 ON "DBA"."Pojazd"  REFERENCING NEW AS Pojazd  FOR EACH ROW  BEGIN  IF Pojazd.Cena > 20000 AND datediff(year, Pojazd.RokProdukcji, GETDATE()) > 20 THEN  RAISERROR 17001 'pojazd nie ma prawidlowej ceny';  ENDIF;  END |

Obraz zawierający tekst

Opis wygenerowany automatycznie

## 8.4 DuzaIloscNapraw

Ten trigger usuwa zarejestrowanie pojazdu gdy ma on za duzo napraw.

Kod:

|  |
| --- |
| ALTER TRIGGER "DuzaIloscNapraw" AFTER INSERT, UPDATE  ORDER 1 ON "DBA"."Naprawy"  REFERENCING NEW AS Napr  FOR EACH ROW  BEGIN  DECLARE ilosc INTEGER;  SET ilosc = (SELECT TOP 1 Count(Naprawy.VIN)  FROM Naprawy  GROUP BY Naprawy.VIN  ORDER BY Count(Naprawy.VIN) DESC);    IF ilosc > 4 THEN  DELETE FROM Rejestracja WHERE VIN=Napr.VIN;    END IF;  END |

Tabela przed:

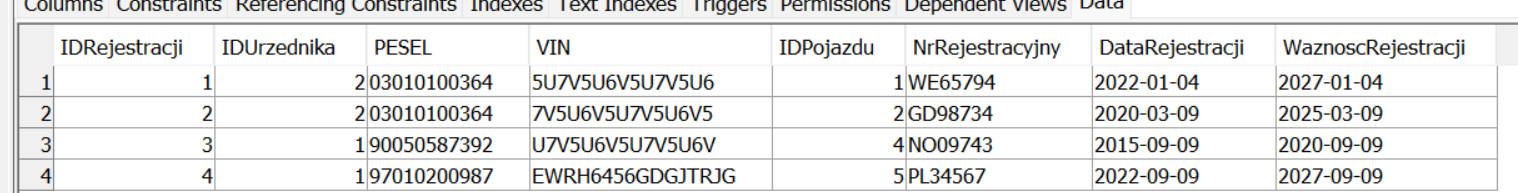
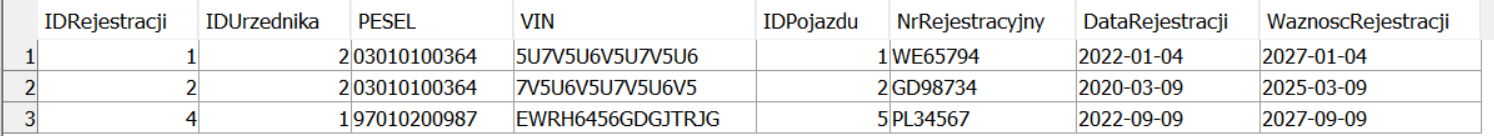


Tabela po:



# 9. Procedury

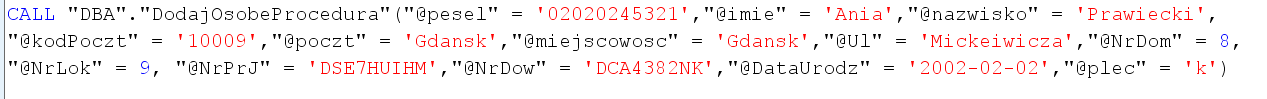
## 9.1. DodajOsobeProcedura

DodajOsobeProcedura to procedura z transakcją, która dodaje właściciela gdy wiek jest większy niż 15 lat. W Innym przypadku transakcja jest cofana

Kod:

|  |
| --- |
| ALTER PROCEDURE "DBA"."DodajOsobeProcedura"(IN pesel CHAR(11), imie VARCHAR(15),  nazwisko VARCHAR(15), kodPoczt CHAR(5), poczt VARCHAR(15), miejscowosc VARCHAR(15),  Ul VARCHAR(15), NrDom INTEGER, NrLok INTEGER, NrPrJ VARCHAR(15), NrDow VARCHAR(9), DataUrodz DATE, plec CHAR(1))  AS  BEGIN  BEGIN TRANSACTION Dodawanie  INSERT INTO Osoba  VALUES(pesel,imie,nazwisko,kodPoczt, poczt,miejscowosc,Ul, NrDom, NrLok,  NrPrJ, NrDow,DataUrodz,plec)  IF datediff(year, dataUrodz, GETDATE()) < 15  BEGIN  ROLLBACK TRANSACTION Dodawanie  END  IF datediff(year, dataUrodz, GETDATE()) >= 15  BEGIN  COMMIT TRANSACTION Dodawanie  END  END |

Wywołanie:



Rezultat:



## 9.2. WyswietlPojazdyProcedura

Jest to procedura z kursorem, która wyświetla wszystkie pojazdy danej osoby oraz województwo z jakiego jest samochód za pomocą innej funkcji.

Kod:

|  |
| --- |
| ALTER PROCEDURE "DBA"."WyswietlPojazdyProcedura"( IN pesel1 VARCHAR(11) )  BEGIN  DECLARE wojewodztwo CHAR(30);  FOR petla1 AS kursor1 CURSOR FOR  (SELECT Osoba.PESEL, Osoba.Nazwisko, Rejestracja.VIN,  Rejestracja.NrRejestracyjny, Marka.NazwaMarki  FROM Osoba  INNER JOIN Rejestracja ON Osoba.PESEL=Rejestracja.PESEL  INNER JOIN Pojazd ON Pojazd.VIN = Rejestracja.VIN  INNER JOIN Marka ON Pojazd.IDMarki=Marka.IDMarki  WHERE Osoba.PESEL=pesel1)  DO  MESSAGE 'Nazwisko: '||Nazwisko||' VIN: '||VIN||' Nr Rejestracyjny: '||NrRejestracyjny||' Marka: '||NazwaMarki TO CLIENT ;  MESSAGE 'Wojewodztwo: '||WojewodztwoFunkcja(NrRejestracyjny) TO CLIENT;  END FOR;  END |

Wywołanie:



Rezultat:

Obraz zawierający tekst

Opis wygenerowany automatycznie

## 9.3. PrzegladProcedura

Jest to Procedura która w przypadku przekazania VIN bada czy jest jakiś pojazd sprzed 2000 roku i w razie spełnienia warunków wysyła go na przegląd.

Kod:

|  |
| --- |
| ALTER PROCEDURE "DBA"."PrzegladProcedura"(VINN VARCHAR(17))  AS  BEGIN  DECLARE @RokProd CHAR(4)    SET @RokProd = (SELECT Pojazd.RokProdukcji FROM Pojazd WHERE Pojazd.VIN=VINN)  IF @RokProd < 2000  BEGIN  DECLARE @nrrej VARCHAR(8)  DECLARE @idpoj INTEGER  SET @nrrej = (SELECT NrRejestracyjny FROM Pojazd WHERE VIN=VINN)  SET @idpoj = (SELECT IDPojazdu FROM Pojazd WHERE VIN=VINN)    UPDATE Naprawy  SET Naprawy.CzescSamochodu = 'przeglad',Naprawy.DataNaprawy = GETDATE() WHERE Naprawy.VIN = VINN  END  IF @RokProd >= 2000  MESSAGE 'jeszcze nie czas na przeglad' TO CLIENT  END |

Wywołanie:



Rezultat:



## 9.4. SprawdzaniePeseluProcedura

Jest to procedura sprawdzająca poprawność wpisanej daty urodzenia za pomocą peselu.

Kod:

|  |
| --- |
| ALTER PROCEDURE "DBA"."SprawdzaniePeseluProcedura"(IN pesel1 char(11))  BEGIN  DECLARE rok INTEGER;  DECLARE miesiac INTEGER;  DECLARE dzien INTEGER;  DECLARE dataUr DATE;  DECLARE dataPESEL CHAR(8);  DECLARE dataPESELdata DATE;    SET dataUr=(SELECT DataUrodzenia FROM Osoba WHERE PESEL=pesel1);  SET rok=LEFT(pesel1, 2);  SET miesiac=RIGHT(LEFT(pesel1, 4),2);  SET dzien=RIGHT(LEFT(pesel1, 6),2);    IF miesiac>20 THEN  SET rok=rok+2000;  SET miesiac=miesiac-20;  ELSE  SET rok=rok+1900;  END IF;  IF miesiac<10 AND dzien>10 THEN  SET dataPESEL=rok||'0'||miesiac||dzien;  END IF;  IF miesiac<10 AND dzien<10 THEN  SET dataPESEL=rok||'0'||miesiac||'0'||dzien;  END IF;  IF miesiac>10 AND dzien<10 THEN  SET dataPESEL=rok||miesiac||'0'||dzien;  END IF;  SET dataPESELdata = CONVERT(date,dataPESEL);  IF dataPESELdata<>DataUr THEN  UPDATE Osoba  SET DataUrodzenia=dataPESELdata  WHERE PESEL=pesel1;  END IF;  END |

Wywołanie:



Przed:



Po:



## 9.5. SprawdzaniePlciProcedura

Jest to procedura sprawdzająca płeć za pomocą ostatniej literki imienia.

Kod:

|  |
| --- |
| ALTER PROCEDURE "DBA"."SprawdzaniePlciProcedura"(IN pesel1 CHAR(11))  BEGIN  DECLARE plecc CHAR(1);  DECLARE plecprawidlowa CHAR(1);  DECLARE imie1 CHAR(15);  SET imie1 = (SELECT Imie FROM Osoba WHERE PESEL=pesel1);  SET plecc = (SELECT Plec FROM Osoba WHERE PESEL=pesel1);  IF imie1 LIKE '%a' THEN SET plecprawidlowa = 'k';  END IF;  IF imie1 NOT LIKE '%a' THEN SET plecprawidlowa = 'm';  END IF;  IF plecc <> plecprawidlowa THEN  UPDATE Osoba  SET Plec=plecprawidlowa  WHERE PESEL = pesel1;  END IF;  END |

Wywołanie:



Przed:



Po:



# 10. Funkcje

## 10.1. WojewodztwoFunkcja

Jest to funkcja zwracająca do jakiego województwa należy rejestracja przekazana w argumencie

Kod:

|  |
| --- |
| ALTER FUNCTION "DBA"."WojewodztwoFunkcja"( IN NrRejestracyjny VARCHAR(8) )  RETURNS VARCHAR(50)  NOT DETERMINISTIC  BEGIN  DECLARE "wojewodztwo" VARCHAR(50);  CASE  WHEN NrRejestracyjny LIKE 'Z%' THEN SET wojewodztwo='wojewodztwo zachodniopomorskie'  WHEN NrRejestracyjny LIKE 'G%' THEN SET wojewodztwo='wojewodztwo pomorskie'  WHEN NrRejestracyjny LIKE 'N%' THEN SET wojewodztwo='wojewodztwo warminsko-mazurskie'  WHEN NrRejestracyjny LIKE 'B%' THEN SET wojewodztwo='wojewodztwo podlaskie'  WHEN NrRejestracyjny LIKE 'F%' THEN SET wojewodztwo='wojewodztwo lubuskie'  WHEN NrRejestracyjny LIKE 'P%' THEN SET wojewodztwo='wojewodztwo wielkopolskie'  WHEN NrRejestracyjny LIKE 'W%' THEN SET wojewodztwo='wojewodztwo mazowieckie'  WHEN NrRejestracyjny LIKE 'L%' THEN SET wojewodztwo='wojewodztwo lubelskie'  WHEN NrRejestracyjny LIKE 'E%' THEN SET wojewodztwo='wojewodztwo lodzkie'  WHEN NrRejestracyjny LIKE 'S%' THEN SET wojewodztwo='wojewodztwo slaskie'  WHEN NrRejestracyjny LIKE 'D%' THEN SET wojewodztwo='wojewodztwo dolnoslaskie'  WHEN NrRejestracyjny LIKE 'K%' THEN SET wojewodztwo='wojewodztwo malopolskie'  WHEN NrRejestracyjny LIKE 'R%' THEN SET wojewodztwo='wojewodztwo podkarpackie'  WHEN NrRejestracyjny LIKE 'T%' THEN SET wojewodztwo='wojewodztwo swietokrzyskie'  WHEN NrRejestracyjny LIKE 'O%' THEN SET wojewodztwo='wojewodztwo opolskie'  WHEN NrRejestracyjny LIKE 'C%' THEN SET wojewodztwo='wojewodztwo kujawsko-pomorskie'  WHEN NrRejestracyjny LIKE 'H%' THEN SET wojewodztwo='sluzby specialne'    ELSE SET wojewodztwo='blad w rejestracji'  END;  RETURN "wojewodztwo";  END |

Wywołanie:



Rezultat:

Obraz zawierający tekst

Opis wygenerowany automatycznie

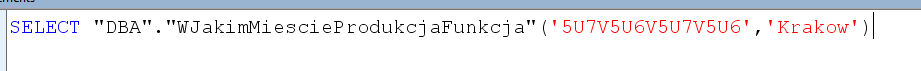
## 10.2. WJakimMiescieProdukcjaFunkcja

Ta funkcja określa czy pojazd o podanym VIN został wyprodukowany w podanej miejscowości, jest to funkcja używająca kursora

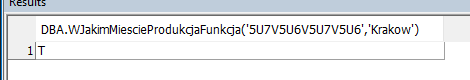
Kod:

|  |
| --- |
| ALTER FUNCTION "DBA"."WJakimMiescieProdukcjaFunkcja"(IN vin1 CHAR(17), miejscowosctest CHAR(15))  RETURNS CHAR(1)//zwraca czy tam zostal wykonany pojazd  DETERMINISTIC  BEGIN  DECLARE czyTak CHAR(1);  DECLARE miejscowosc1 CHAR(15);  DECLARE kursor CURSOR FOR  (SELECT Fabryka.Miejscowosc FROM Fabryka  INNER JOIN Pojazd ON Fabryka.IDFabryki = Pojazd.IDFabryki  INNER JOIN Rejestracja ON Rejestracja.VIN=Pojazd.VIN  WHERE Pojazd.VIN = vin1);  SET czyTak = 'F';  OPEN kursor;  petla1: LOOP  FETCH NEXT kursor INTO miejscowosc1;  IF SQLCODE <> 0 THEN  LEAVE petla1;  ENDIF;  IF miejscowosc1 = miejscowosctest THEN  SET czyTak = 'T';  LEAVE petla1;  ENDIF;  END LOOP;  CLOSE kursor;  RETURN czyTak;  END |

Wywołanie:



Rezultat:



Przykład 2

Wywołanie:



Rezultat:



## 10.3. WaznoscRejestracjiFunkcja

Jest to funkcja która bada czy data ważności jest poprawna i w razie potrzeby aktualizuje ją. Zwraca informację o aktualizacji.

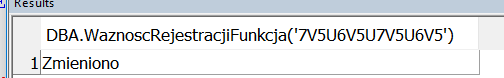
Kod:

|  |
| --- |
| ALTER FUNCTION "DBA"."WaznoscRejestracjiFunkcja"(IN NrVIN VARCHAR(17))  RETURNS CHAR(10)  NOT DETERMINISTIC  BEGIN  DECLARE CzyZmieniono CHAR(10);  DECLARE dataRej DATE;  DECLARE dataWaz DATE;  DECLARE dataWazZTab DATE;  SET dataRej = (SELECT Rejestracja.DataRejestracji FROM Rejestracja WHERE Rejestracja.VIN=NrVIN);  SET dataWaz = DATEADD(YEAR,5,dataRej);  SET dataWazZTab = (SELECT Rejestracja.WaznoscRejestracji FROM Rejestracja WHERE Rejestracja.VIN=NrVIN);  IF dataWazZTab=dataWaz THEN  SET CzyZmieniono='Nie zmieniono';  ENDIF;  IF dataWazZTab<>dataWaz THEN  UPDATE Rejestracja  SET Rejestracja.WaznoscRejestracji = dataWaz WHERE Rejestracja.VIN=NrVIN;  SET CzyZmieniono='Zmieniono';  ENDIF;  RETURN CzyZmieniono;  END |

Wywołanie:



Rezultat:





## 10.4. WadliwaMarkaFunkcja

Jest to funkcja, która zwraca ID Marki która ma najwięcej zarejestrowanych napraw.

Kod:

|  |
| --- |
| ALTER FUNCTION "DBA"."WadliwaMarkaFunkcja"()  RETURNS INTEGER  BEGIN  DECLARE @marka INTEGER;  SET @marka=  (SELECT TOP 1 Marka.IDMarki  FROM Marka  INNER JOIN Pojazd ON Pojazd.IDMarki=Marka.IDMarki  INNER JOIN Naprawy ON Naprawy.IDPojazdu = Pojazd.IDPojazdu  GROUP BY Marka.IDMarki  ORDER BY COUNT(Pojazd.IDMarki) DESC);    RETURN @marka;  END |

Wywołanie:



Rezultat:



## 10.5. UsunFabrykeFunkcja

Jest to funkcja wykorzystująca narzędzie transakcji. Transakcja usuwa fabrykę, ale tylko jeśli nie wyprodukowała żadnego samochodu.

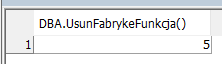
Kod:

|  |
| --- |
| ALTER FUNCTION "DBA"."UsunFabrykeFunkcja"()  RETURNS INTEGER  AS  BEGIN  BEGIN TRANSACTION Usuniecie  DECLARE @ileWyprodukowano INTEGER  DECLARE @idF INTEGER  SET @ileWyprodukowano =(SELECT TOP 1 COALESCE( COUNT(Pojazd.IDFabryki), 0) AS Ilosc  FROM Fabryka LEFT JOIN Pojazd ON Fabryka.IDFabryki=Pojazd.IDFabryki  GROUP BY Fabryka.IDFabryki  ORDER BY Ilosc)  SET @idF = (SELECT TOP 1 Fabryka.IDFabryki  FROM Fabryka LEFT JOIN Pojazd ON Fabryka.IDFabryki=Pojazd.IDFabryki  GROUP BY Fabryka.IDFabryki  HAVING COALESCE(COUNT(Pojazd.IDFabryki), 0) = 0)  DELETE FROM Fabryka WHERE IDFabryki=@idF    UPDATE Pojazd  SET IDFabryki=0 WHERE IDFabryki=@idF  IF @ileWyprodukowano <> 0  BEGIN  ROLLBACK TRANSACTION Usuniecie  END  IF @ileWyprodukowano = 0  BEGIN  COMMIT TRANSACTION Usuniecie  END  RETURN @idF  END |

Wywołanie:



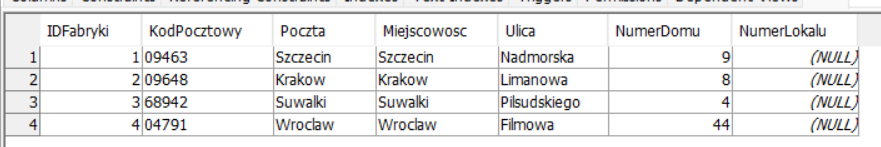
Rezultaty:



Przed:



Po:



# 11. Użytkownicy

Utworzono trzech użytkowników:

## 11.1 WlascicielPojazdu

Ten użytkownik ma dostęp do zasobów, funkcji i procedur związanych z pojazdami, danymi o pojeździe oraz do swoich danych.

## 11.2. Urzednik

Ten użytkownik ma dostęp do informacji o rejestracji i pojeździe, a także do danych Urzędów.

## 11.3. DealerSamochodow

Ten użytkownik ma dostęp do danych o pojazdach, ich wyposażeniu, naprawach i fabrykach.

## 11.4. Zestawienie uprawnień

|  |  |  |  |
| --- | --- | --- | --- |
|  | **WlascicielPojazdu** | **Urzednik** | **DealerSamochodow** |
| Osoba | **✓** | **✓** |  |
| Rejestracja | **✓** | **✓** |  |
| Urzednik |  | **✓** |  |
| Urzad |  | **✓** |  |
| Pojazd | **✓** | **✓** | **✓** |
| RodzajWyposazenia | **✓** |  | **✓** |
| DodatkoweWyposazenie |  |  | **✓** |
| Naprawy | **✓** |  | **✓** |
| Fabryka |  |  | **✓** |
| Marka |  |  | **✓** |
| InfoOPojezdzie | **✓** |  | **✓** |
| UrzednikW |  | **✓** |  |
| FabrykaW |  |  | **✓** |
| KtoIleWydalW | **✓** |  | **✓** |
| DodajOsobeProcedura | **✓** | **✓** |  |
| WyswietlPojazdyProcedura | **✓** | **✓** |  |
| PrzegladProcedura | **✓** |  | **✓** |
| SprawdzeniePeseluProcedura |  | **✓** |  |
| SprawdzaniePlciProcedura |  | **✓** |  |
| WojewodztwoFunkcja |  |  |  |
| WJakimMiescieProdukcjaFunkcja |  |  | **✓** |
| WaznoscRejestracjiFunkcja | **✓** | **✓** |  |
| WadliwaMarkaFunkcja |  |  | **✓** |
| UsunFabrykeFunkcja |  |  | **✓** |

W tabeli Marka użyto mechanizmu autoincrement

