

WOJSKOWA AKADEMIA TECHNICZNA

Laboratorium z przedmiotu

BAZY DANYCH

TEMAT: OBSŁUGA ZAMÓWIEŃ W SALONACH
SAMOCHODOWYCH

Wykonał: Konrad Kosiński

Grupa: I*Bd D

Założenia projektu

Baza danych stworzona została z myślą o sieci salonów samochodowych składającej się z trzech salonów sprzedających samochody i akcesoria do samochodów. Wszystkie salony znajdują się w Polsce, jeden w Warszawie, drugi w Białymstoku a trzeci w Łodzi. Każdy pracownik jest przypisany do jednego salonu, natomiast klient może kupić towary w dowolnym salonie, faktury klienta ze wszystkich salonów przechowywane są w bazie danych. Co więcej, w bazie przechowywane są także dane na temat towarów oraz producentów tych towarów. Każdy towar ma przypisanego jednego producenta oraz każde akcesoria mają przypisany jeden typ towaru np. "wycieraczki".

Opis poszczególnych encji

SalonSamochodowy - jest to lokal, w którym klient może kupić towary(auta lub akcesoria do nich), w bazie znajdują się trzy salony.

Samochód - jest to samochód w salonie przeznaczony do sprzedaży

Akcesoria - są to akcesoria przydatne do samochodów takie jak wycieraczki, opony czy nawigacja przeznaczone do sprzedaży w salonie.

Pracownik - jest to osoba pracująca w salonie, przypisana może być tylko do jednego salonu

Typ - jest to dana kategoria Akcesorii, np. wycieraczki, opony.

Klient - jest to osoba kupująca towary w salonie.

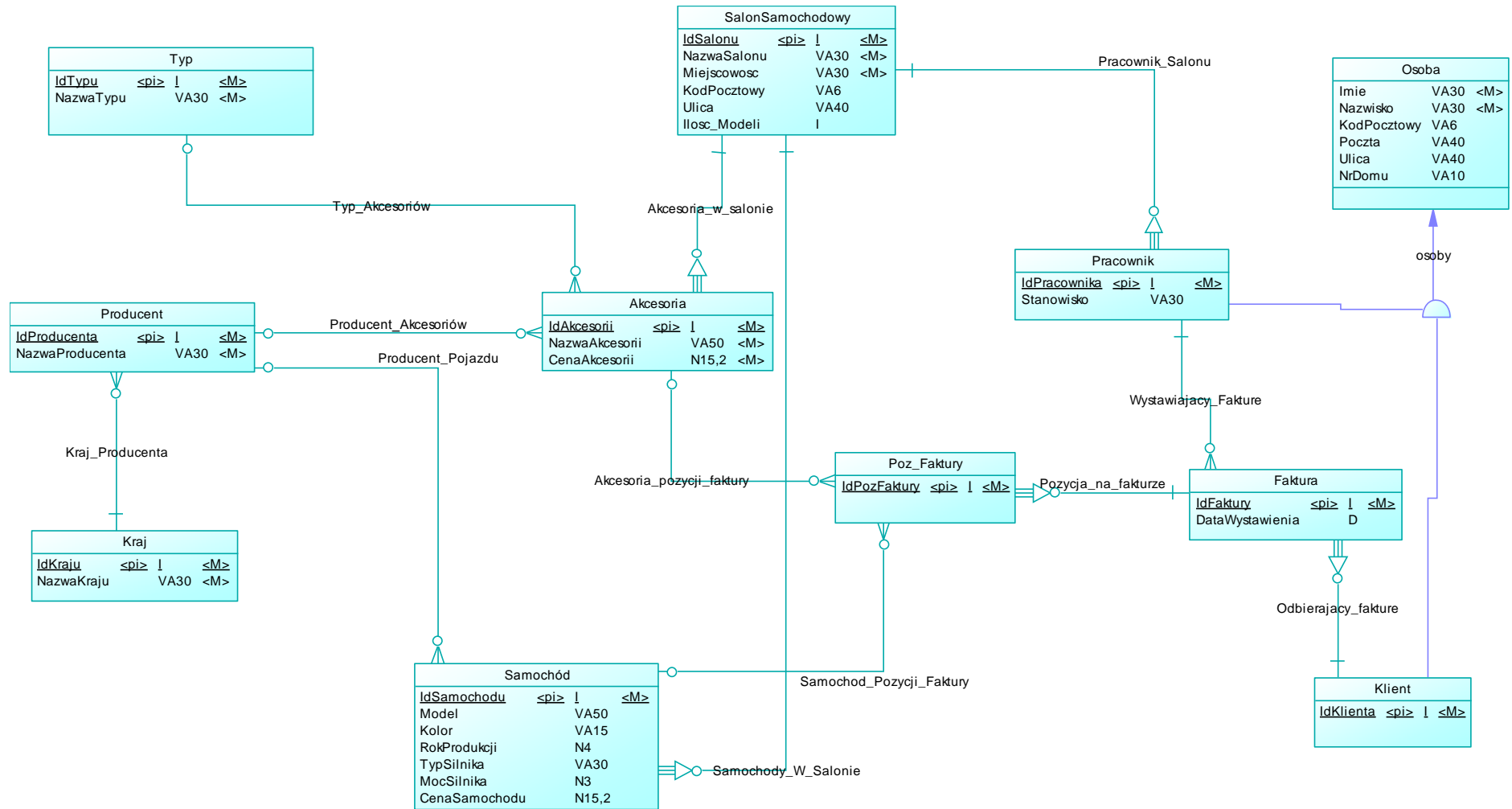
Faktura - jest to dokument zakupu towaru lub towarów w salonie.

Poz_Faktury - jest to pozycja oznaczająca jeden z towarów lub akcesorii zakupionych na danej fakturze

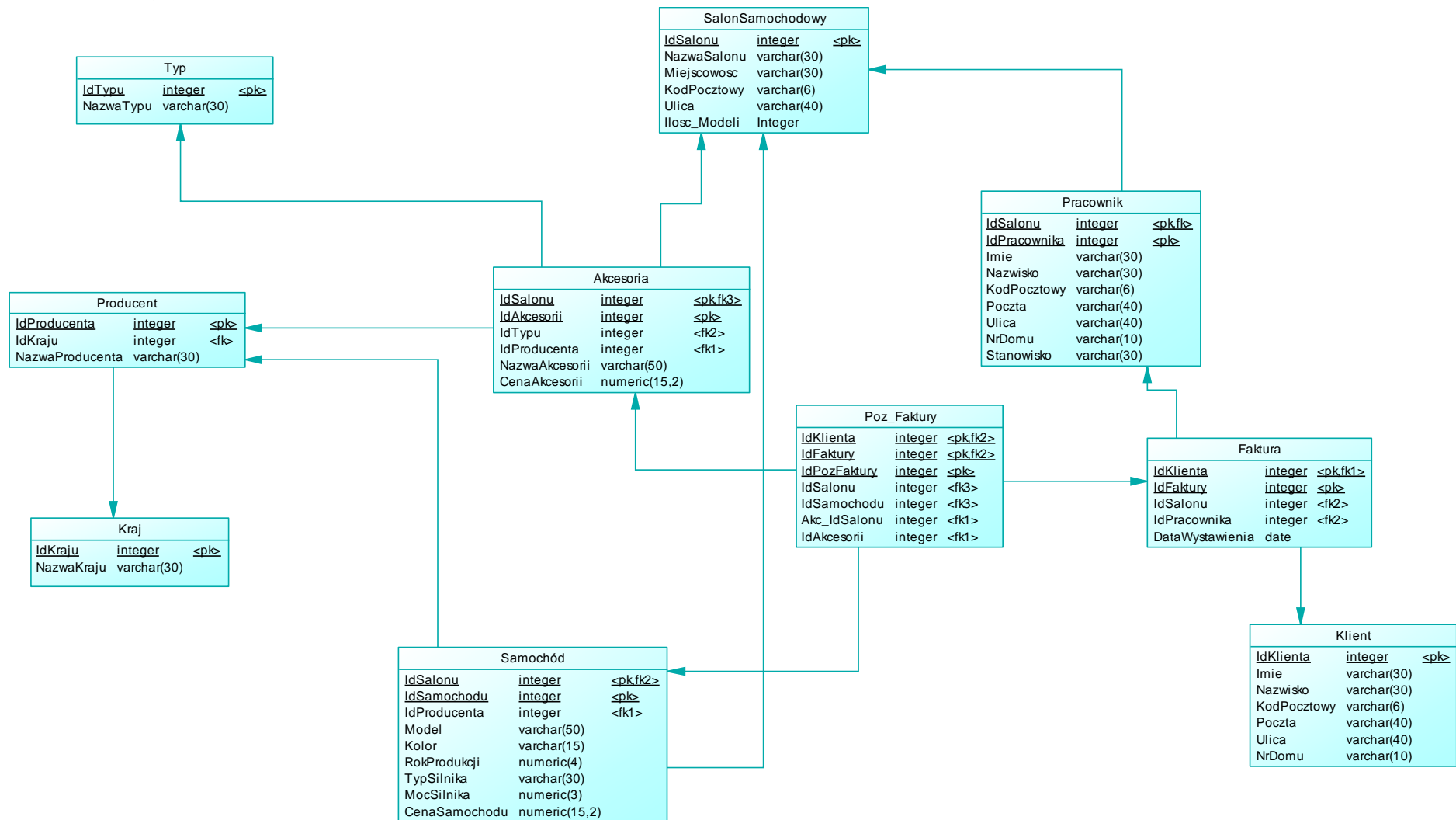
Producent – jest to firma produkująca samochody lub akcesoria.

Kraj – jest to państwo, z którego pochodzi dany producent

1. Model konceptualny



2. Model fizyczny



3.WIDOKI

a) lista producentów samochodów z krajem pochodzenia producenta oraz z liczbą sprzedanych aut, uporządkowana malejąco według łącznej kwoty sprzedanych aut tego producenta

```
CREATE or REPLACE VIEW "DBA"."bestCarProducers"()
AS
SELECT P.NazwaProducenta, K.NazwaKraju,
sum(S.CenaSamochodu) AS LacznyPrzychod,
count(S.IdSamochodu) AS LiczbaSprzedanych
FROM Producent P
INNER JOIN Kraj K ON P.IdKraju=K.IdKraju
INNER JOIN Samochód S ON S.IdProducenta=P.IdProducenta
INNER JOIN Poz_Faktury PF ON PF.IdSamochodu=S.IdSamochodu
AND PF.IdSalonu=S.IdSalonu
GROUP BY P.NazwaProducenta, K.NazwaKraju
ORDER BY LacznyPrzychod DESC
```

b) lista klientów z liczbą kupionych samochodów uporządkowanych malejąco według łącznej kwoty zakupionych przedmiotów

```
CREATE VIEW "DBA"."bestCarCustomers"()
AS
SELECT K.Nazwisko, K.Imie, count(S.IdSamochodu)
AS LiczbaKupionychS, sum(S.CenaSamochodu) AS KwotaKupionychS
FROM Klient K
INNER JOIN Faktura F ON K.IdKlienta=F.IdKlienta
INNER JOIN Poz_Faktury PF ON (F.IdFaktury=PF.IdFaktury
AND PF.IdSalonu=F.IdSalonu
AND PF.IdKlienta=F.IdKlienta)
Inner JOIN Samochód S on S.IdSamochodu=PF.IdSamochodu
AND S.IdSalonu=PF.IdSalonu
GROUP BY K.Nazwisko, K.Imie
ORDER BY KwotaKupionychS DESC
```

c) lista najlepiej sprzedających się akcesoriów we wszystkich salonach uporządkowana liczbą sprzedanych produktów malejąco

```
CREATE VIEW "DBA"."AkcesoriaBestSeller"()
AS
SELECT TY.NazwaTypu, A.NazwaAkcesorii,
count(A.IdAkcesorii) AS LiczbaSprzedanych,
sum(A.CenaAkcesorii) AS LacznyPrzychod
FROM Akcesoria A
INNER JOIN Poz_Faktury PF ON PF.IdAkcesorii=A.IdAkcesorii
AND PF.Akc_IdSalonu=A.IdSalonu
INNER JOIN Typ TY ON A.IdTypu=TY.IdTypu
GROUP BY A.NazwaAkcesorii, TY.NazwaTypu
ORDER BY LiczbaSprzedanych DESC
```

d) Widok Zmaterializowany odświeżany w procedurze nowyA. Lista salonów wyświetlająca ID salonu, nazwę salonu i łączną wartość akcesoriów znajdujących się w danym Salonie. Lista uporządkowana łączną wartością akcesoriów malejąco.

```
CREATE MATERIALIZED VIEW "DBA"."AkcesoriaTotalValue"
AS
SELECT S.IdSalonu,S.NazwaSalonu,Sum(A.CenaAkcesorii) as Laczna_Wartosc
FROM SalonSamochodowy S INNER JOIN Akcesoria A
on S.IdSalonu = A.IdSalonu
GROUP BY S.IdSalonu,S.NazwaSalonu
ORDER BY Laczna_Wartosc DESC
```

4. Triggery

a.) zapobiega usunięciu producenta jeśli jest jakiś jego samochód w bazie

```
CREATE OR REPLACE TRIGGER "TRIG_Producent" BEFORE DELETE
ORDER 1 ON "DBA"."Producent"
REFERENCING OLD AS old_name NEW AS new_name
FOR EACH ROW
BEGIN
IF(EXISTS(SELECT P.IdProducenta FROM Producent P
          INNER JOIN Samochód S ON S.IdProducenta=P.IdProducenta
          WHERE P.IdProducenta=new_name.IdProducenta))
THEN
    RAISERROR 30002 'Nie mozna usunac producenta gdy znajduje sie
samochód tego producenta w bazie!';
    MESSAGE 'Nie mozna usunac producenta gdy znajduje sie samochód tego
producenta w bazie!' TO client
END IF
END;
```

b.) zapobiega usunięciu typu jeśli w bazie znajduje sie akcesoria tego typu

```
CREATE OR REPLACE TRIGGER "TRIG_Typ" BEFORE DELETE
ORDER 1 ON "DBA"."Typ"
REFERENCING OLD AS old_name NEW AS new_name
FOR EACH ROW
BEGIN
IF(EXISTS(SELECT TY.IdTypu FROM Typ TY
          INNER JOIN Akcesoria A ON A.IdTypu=TY.IdTypu
          WHERE TY.IdTypu=new_name.IdTypu))
THEN
    RAISERROR 30002 'Nie mozna usunac typu gdy znajduje sie akcesoria tego
typu w bazie!';
    MESSAGE 'Nie mozna usunac typu gdy znajduje sie akcesoria tego typu w
bazie!' TO client
END IF
END;
```

c.) zapobiega usunięciu samochodu jeśli jakiś znajduje się na pozycji faktury w bazie.

```
CREATE OR REPLACE TRIGGER "TRIG_Samochod" BEFORE DELETE
ORDER 1 ON "DBA"."Samochód"
REFERENCING OLD AS old_name NEW AS new_name
FOR EACH ROW
BEGIN
IF(EXISTS(SELECT P.IdPozFaktury FROM Poz_Faktury P
          INNER JOIN Samochód S ON S.IdSamochodu=P.IdSamochodu AND
          S.IdSalonu = p.IdSalonu
          WHERE S.IdSamochodu=new_name.IdProducenta))
THEN
    RAISERROR 30002 'Nie mozna usunac samochodu gdyż ten samochód jest
na pozycji faktury!';
    MESSAGE 'Nie mozna usunac samochodu gdyż ten samochód jest na
pozycji faktury!' TO client
END IF
END;
```

d)zapobiega aktualizacji ID samochodu na pozycji faktury gdyż przyjmujemy że faktura już została wystawiona.

```
CREATE OR REPLACE TRIGGER "TRIG_PozFaktury" BEFORE UPDATE
ORDER 1 ON "DBA"."Poz_Faktury"
FOR EACH ROW
BEGIN
                                IF EXISTS (SELECT P.IdPozFaktury FROM
Poz_Faktury P
          INNER JOIN Samochód S ON S.IdSamochodu=P.IdSamochodu AND
          S.IdSalonu = p.IdSalonu
          )
THEN
    RAISERROR 30002 'Nie mozna zmienić IdSamochodu na fakturze gdyż
faktura została wystawiona!';
    MESSAGE 'Samochód sprzedany. Faktura jest już wystawiona!' TO client
END IF
END;
```


e) Trigger ten uruchomi się po wstawieniu nowego wiersza, a także edycji lub usunięciu wiersza z tabeli Samochód. Sprawdzi on jakiego salonu dotyczy zmiana kolumny IdSalonu i w zależności czy zmieniamy przynależność istniejącego salonu dla istniejącego już samochodu, czy dodamy nowe auto lub je usuniemy to wartość kolumny Ilosc_Modeli w tabeli SalonSamochodowy zwiększy się lub zmniejszy.

```
CREATE or REPLACE TRIGGER "TRIG_Salon" AFTER INSERT, DELETE, UPDATE  
ORDER 1 ON "DBA"."Samochód"
```

```
REFERENCING NEW AS PoZmianie OLD AS PrzedZmiana  
FOR EACH ROW
```

```
BEGIN
```

```
DECLARE @IdSalonu int;
```

```
DECLARE @LM1 int;
```

```
DECLARE @LM2 int;
```

```
SET @LM1=(SELECT count(*)  
FROM Samochód S  
WHERE S.IdSalonu=PoZmianie.IdSalonu);
```

```
SET @LM2=(SELECT count(*)  
FROM Samochód S  
WHERE S.IdSalonu=PrzedZmiana.IdSalonu);
```

```
UPDATE SalonSamochodowy SS SET Ilosc_Modeli = @LM1 WHERE  
SS.IdSalonu=PoZmianie.IdSalonu;
```

```
UPDATE SalonSamochodowy SS SET Ilosc_Modeli = @LM2 WHERE  
SS.IdSalonu=PrzedZmiana.IdSalonu;
```

```
END;
```

f) Przed każdą próbą dodania nowego rekordu lub aktualizowania rekordu w tabeli klient sprawdza czy KodPocztowy ma dokładnie 6 znaków.

```
CREATE TRIGGER "TRIG_Klient" BEFORE INSERT, UPDATE  
ORDER 1 ON "DBA"."Klient"
```

```

REFERENCING NEW AS Klient
FOR EACH ROW
BEGIN
declare k integer;
set k = (char_length(Klient.KodPocztowy));
IF k < 6 THEN
SIGNAL za_krotki_KOD_POCZTOWY
end if;
if k > 6 THEN
SIGNAL za_dlugi_KOD_POCZTOWY
end if;
END

```

g) Trigger taki sam jak w punkcie f tylko że dla tabeli Pracownik

```

CREATE TRIGGER "TRIG_Pracownik" BEFORE INSERT, UPDATE
ORDER 1 ON "DBA"."Pracownik"
REFERENCING NEW AS Pracownik
FOR EACH ROW
BEGIN
declare k integer;
set k = (char_length(Pracownik.KodPocztowy));
IF k < 6 THEN
SIGNAL za_krotki_KOD_POCZTOWY
end if;
if k > 6 THEN
SIGNAL za_dlugi_KOD_POCZTOWY
end if;
END

```

5. Funkcje

a.) funkcja dodająca producenta do bazy danych. Zwracany jest identyfikator utworzonego producenta. W funkcji użyto transakcji oraz kursora.

```

CREATE OR REPLACE FUNCTION "DBA"."Kursor_New_Producent"( IN
ProducentNazwa varchar(30), KrajNazwa varchar(30)
/* @parameter_name parameter_type [= default_value], ... */ )
RETURNS BIGINT
AS
BEGIN
DECLARE @idP BIGINT
DECLARE @kursor CURSOR FOR (SELECT NazwaProducenta FROM Producent)
DECLARE @tmp int
DECLARE @producentN varchar(50)
DECLARE @idK int
DECLARE @licznik integer

BEGIN TRAN tra

SET @licznik=1
OPEN kursor
WHILE (@licznik<(SELECT COUNT(*) FROM Producent))
BEGIN
    FETCH NEXT kursor INTO @producentN

    SET @tmp=1

    IF(@producentN=@ProducentNazwa)
        SET @tmp=0

    IF(@producentN=@ProducentNazwa)
        SET @licznik=(SELECT COUNT(*) FROM Producent)

    SET @licznik=@licznik+1
END

SET @idK = (SELECT idKraju FROM Kraj K
            WHERE K.NazwaKraju=KrajNazwa)

```

```
SET @idP=(SELECT max(idProducenta) FROM Producent)
```

```
SET @idP=@idP+1
```

```
INSERT INTO Producent VALUES(@idP, @idK, @ProducentNazwa)
```

```
IF(@tmp=1) COMMIT TRAN tra
```

```
ELSE
```

```
BEGIN
```

```
    ROLLBACK TRAN tra
```

```
    SET @idP=0
```

```
END
```

```
RETURN @idP
```

```
END;
```

b.) funkcja zwracająca przychód dla podanego salonu, gdzie parametrem wejściowym jest identyfikator Salonu

```
CREATE OR REPLACE FUNCTION "DBA"."Salon_Income"( IN SalonID INT)
```

```
RETURNS DOUBLE
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    DECLARE "wynik" DOUBLE;
```

```
    DECLARE "zmienna1" DOUBLE;
```

```
    DECLARE "zmienna2" DOUBLE;
```

```
    SET zmienna1 = (SELECT sum(A.CenaAkcesorii)
```

```
        FROM Poz_Faktury PF JOIN Akcesoria A
```

```
        ON A.IdSalonu=PF.Akc_IdSalonu
```

```
        AND A.IdAkcesorii=PF.IdAkcesorii
```

```
        WHERE PF.Akc_IdSalonu=SalonID);
```

```
    SET zmienna2 = (SELECT sum(S.CenaSamochodu)
```

```
        FROM Poz_Faktury PF JOIN Samochód S
```

```
        ON S.IdSamochodu=PF.IdSamochodu
```

```
        AND S.IdSalonu=PF.IdSalonu
```

```
        WHERE PF.IdSalonu=SalonID);
```

```
    SET wynik = COALESCE(zmienna1 + zmienna2, zmienna1,zmienna2);
```

```
    RETURN "wynik";
```

END;

c.) funkcja zwracająca liczbę wystawionych faktur przez danego pracownika

```
CREATE OR REPLACE FUNCTION "DBA"."Pracownik_HowManyFactures"( IN  
Pracownik_imie varchar(30), Pracownik_nazwisko varchar(30))
```

```
RETURNS BIGINT
```

```
DETERMINISTIC
```

```
BEGIN
```

```
DECLARE "wynik" BIGINT;
```

```
SET wynik= (SELECT COUNT(*) FROM Faktura F  
            INNER JOIN Pracownik P ON F.idSalonu=P.idSalonu  
            AND F.idPracownika=P.idPracownika  
            WHERE P.Imie = Pracownik_imie  
            AND P.Nazwisko = Pracownik_nazwisko);
```

```
RETURN "wynik";
```

END;

d.) funkcja zwracająca liczbę sprzedanych aut danej marki producenta.

```
CREATE OR REPLACE FUNCTION "DBA"."HowManyCars"( in "nazwaMarki"  
varchar(30) )
```

```
RETURNS INTEGER
```

```
DETERMINISTIC
```

```
BEGIN
```

```
DECLARE "ilosc_aut" INTEGER;
```

```
SET ilosc_aut = (SELECT COUNT(*)
```

```
FROM Producent p
```

```
INNER JOIN Samochód s ON s.IdProducenta = p.IdProducenta
```

```
INNER JOIN SalonSamochodowy ss ON ss.IdSalonu = s.IdSalonu
```

```
INNER JOIN Poz_Faktury pf ON pf.IdSamochodu = s.IdSamochodu AND  
pf.IdSalonu = ss.IdSalonu
```

```
WHERE p.NazwaProducenta ="nazwaMarki"
```

```
);
```

```
RETURN "ilosc_aut";
```

END;

6. Procedury

a.) procedura wyświetla akcesoria w danym salonie dla zadanego typu

```

CREATE OR REPLACE PROCEDURE "DBA"."Salon_ShowAkcesoria"( IN typTowaru
varchar(50), SalonID int )
RESULT( nazwa_towaru varchar(50), producent_towaru varchar(50),
cena_towaru double)
BEGIN
SELECT A.NazwaAkcesorii, P.NazwaProducenta, A.CenaAkcesorii
FROM Akcesoria A INNER JOIN Producent P
ON P.IdProducenta=A.IdProducenta
INNER JOIN Typ TY ON TY.IdTypu=A.IdTypu
WHERE TY.NazwaTypu=typTowaru
AND A.IdSalonu=SalonID
END;

```

b.) procedura wyświetlająca wszystkie kupione towary przez danego klienta

```

CREATE OR REPLACE PROCEDURE "DBA"."Klient_Purchased"( IN imieKlienta
varchar(30), nazwiskoKlienta varchar(30))
RESULT( Marka varchar(50), nazwa_towaru varchar(50), cena_towaru double )
BEGIN
SELECT P.NazwaProducenta, A.NazwaAkcesorii, A.CenaAkcesorii
FROM Producent P INNER JOIN Akcesoria A ON
P.IdProducenta=A.IdProducenta
INNER JOIN Poz_Faktury PF
ON PF.IdAkcesorii=A.IdAkcesorii AND PF.Akc_IdSalonu=A.IdSalonu
INNER JOIN Klient K ON K.IdKlienta=PF.IdKlienta
WHERE K.Imie=imieKlienta AND K.Nazwisko=nazwiskoKlienta

UNION

SELECT P.NazwaProducenta, S.Model, S.CenaSamochodu
FROM Producent P INNER JOIN Samochód S on
P.IdProducenta=S.IdProducenta
INNER JOIN Poz_Faktury PF
ON PF.IdSamochodu=S.IdSamochodu AND PF.IdSalonu=S.IdSalonu
INNER JOIN Klient K ON K.IdKlienta=PF.IdKlienta
WHERE K.Imie=imieKlienta AND K.Nazwisko=nazwiskoKlienta
END;

```

c.) procedura dodająca nowe akcesoria do wszystkich salonów w bazie. W procedurze użyto transakcji oraz kursora. W procedurze odświeżany jest widok zmaterializowany łączna_wartosc_akcesoriow.

```
CREATE PROCEDURE "DBA"."Kursor_New_Akcesoria"( IN ProducentID int, typID  
int, AkcesoriaNazwa varchar(50) , AkcesoriaCena double  
/* [IN | OUT | INOUT] parameter_name parameter_type [DEFAULT  
default_value], ... */) */  
/* RESULT( column_name column_type, ... ) */  
BEGIN
```

```
                                DECLARE kursor CURSOR FOR (SELECT  
NazwaAkcesorii FROM Akcesoria);
```

```
    DECLARE kursorS CURSOR FOR (SELECT IdSalonu FROM SalonSamochodowy);
```

```
    DECLARE idS int;
```

```
    DECLARE licznik1 int;
```

```
    DECLARE nAkcesorii varchar(100);
```

```
    DECLARE tmp int;
```

```
    DECLARE nrA int;
```

```
    DECLARE licznik2 int;
```

```
    SET licznik1=1;
```

```
    SET licznik2=1;
```

```
    SET tmp=0;
```

```
    OPEN kursor;
```

```
    petla: LOOP
```

```
        FETCH NEXT kursor INTO nAkcesorii;
```

```
        SET licznik2=licznik2+1;
```

```
        SET tmp=0;
```

```
        IF(nAkcesorii=AkcesoriaNazwa) THEN LEAVE petla;
```

```
        ELSE
```

```
            SET tmp=1;
```

```
        END IF;
```

```
        IF(licznik2>(SELECT COUNT(*) FROM Akcesoria)) THEN LEAVE petla;
```

```

    END IF;
END LOOP;
CLOSE kursor;

SET nrA=(SELECT max(IdAkcesorii) FROM Akcesoria);
SET nrA= nrA + 1;

OPEN kursorS;
petla2: LOOP
    FETCH NEXT kursorS INTO idS;
    INSERT INTO Akcesoria(IdAkcesorii, IdProducenta, IdTypu, NazwaAkcesorii,
CenaAkcesorii, IdSalonu)
        VALUES(nrA, ProducentID, typID, AkcesoriaNazwa , AkcesoriaCena,
licznik1);
    SET licznik1=licznik1+1;
    IF(licznik1>(SELECT count(*) FROM SalonSamochodowy)) THEN LEAVE
petla2;
    END IF;
END LOOP;
CLOSE kursorS;

IF(tmp=1) THEN
    COMMIT;
    MESSAGE 'Towar dodany do bazy danych.' TO client
ELSE
    ROLLBACK;
    MESSAGE 'towar NIE dodany do bazy danych, poniewaz inny towar o takiej
samej nazwie znajduje sie w bazie!' TO client
    END IF;
REFRESH MATERIALIZED VIEW AkcesoriaTotalValue;
END;

```

d.) procedura wyświetlająca wszystkie sprzedane samochody w danym salonie.

```

CREATE OR REPLACE PROCEDURE "DBA"."CarsSold"( SalonID int )
RESULT( IdSalonu int, nazwa_salonu varchar(50),

```



```

marka_samochodu varchar(50),model_samochodu varchar(50),
cena_samochodu double)
BEGIN
SELECT SS.IdSalonu, SS.NazwaSalonu, P.NazwaProducenta, S.Model,
S.CenaSamochodu
FROM Samochód S INNER JOIN Producent P
ON P.IdProducenta=S.IdProducenta
INNER JOIN SalonSamochodowy SS ON SS.IdSalonu=S.IdSalonu
INNER JOIN Poz_Faktury pf ON pf.IdSamochodu = S.IdSamochodu
AND pf.IdSalonu = SS.IdSalonu
WHERE SS.IdSalonu=SalonID
END;

```

7. Użytkownicy

a.) użytkownik DBA

Administrator, który ma pełny dostęp do wszystkich operacji na bazie danych

b.) użytkownik Pracownik

Jest to konto pracownika salonu, na którym może edytować oraz dodawać nowe rekordy w bazie danych

c.) użytkownik Klient

Jest to konto Klienta, na którym można przeglądać takie tabele w bazie danych jak Akcesoria, Samochód, SalonSamochodowy, Typ i Producent.

8. Skrypt tworzący bazę danych.

```

/*=====*
/
/* DBMS name:   SAP SQL Anywhere 16          */

```

```

/* Created on: 2017-01-19 15:13:59 */
/*=====*/
/
if exists(select 1 from sys.syssequence s
  where sequence_name='SEQ_PRODUCENT') then
  drop sequence SEQ_PRODUCENT
end if;
if exists(select 1 from sys.syssequence s
  where sequence_name='SEQ_Kraj') then
  drop sequence SEQ_Kraj
end if;
DROP VIEW if exists "DBA"."bestCarCustomers";
DROP VIEW if exists "DBA"."bestCarProducers";
DROP VIEW if exists "DBA"."AkcesoriabestSeller";
DROP MATERIALIZED VIEW if exists "DBA"."AkcesoriaTotalValue"
DROP TRIGGER if exists "TRIG_Klient";
DROP TRIGGER if exists "TRIG_PozFaktury";
DROP TRIGGER if exists "TRIG_Pracownik";
DROP TRIGGER if exists "TRIG_Producent";
DROP TRIGGER if exists "TRIG_Salon"
DROP TRIGGER if exists "TRIG_Samochod"
DROP TRIGGER if exists "TRIG_Typ"
DROP PROCEDURE if exists "DBA"."CarsSold";
DROP PROCEDURE if exists "DBA"."Kursor_New_Akcesoria";
DROP PROCEDURE if exists "DBA"."Klient_Purchased";
DROP PROCEDURE if exists "DBA"."Salon_ShowAkcesoria";
DROP FUNCTION if exists "DBA"."HowManyCars";
DROP FUNCTION if exists "DBA"."Kursor_New_Producent";
DROP FUNCTION if exists "DBA"."Salon_Income";
DROP FUNCTION if exists "DBA"."Pracownik_HowManyFactures";
if exists(select 1 from sys.sysforeignkey where
  role='FK_AKCESORI_AKCESORIA_SALONSAM') then
  alter table Akcesoria
    delete foreign key FK_AKCESORI_AKCESORIA_SALONSAM
end if;

```

```
if exists(select 1 from sys.sysforeignkey where
role='FK_AKCESORI_PRODUCENT_PRODUCEN') then
    alter table Akcesoria
        delete foreign key FK_AKCESORI_PRODUCENT_PRODUCEN
end if;
```

```
if exists(select 1 from sys.sysforeignkey where
role='FK_AKCESORI_TYP_AKCES_TYP') then
    alter table Akcesoria
        delete foreign key FK_AKCESORI_TYP_AKCES_TYP
end if;
```

```
if exists(select 1 from sys.sysforeignkey where
role='FK_FAKTURA_ODBIERAJA_KLIENT') then
    alter table Faktura
        delete foreign key FK_FAKTURA_ODBIERAJA_KLIENT
end if;
```

```
if exists(select 1 from sys.sysforeignkey where
role='FK_FAKTURA_WYSTAWIAJ_PRACOWNI') then
    alter table Faktura
        delete foreign key FK_FAKTURA_WYSTAWIAJ_PRACOWNI
end if;
```

```
if exists(select 1 from sys.sysforeignkey where
role='FK_POZ_FAKT_AKCESORIA_AKCESORI') then
    alter table Poz_Faktury
        delete foreign key FK_POZ_FAKT_AKCESORIA_AKCESORI
end if;
```

```
if exists(select 1 from sys.sysforeignkey where
role='FK_POZ_FAKT_POZYCJA_N_FAKTURA') then
    alter table Poz_Faktury
        delete foreign key FK_POZ_FAKT_POZYCJA_N_FAKTURA
```

end if;

*if exists(select 1 from sys.sysforeignkey where
role='FK_POZ_FAKT_SAMOCHOD__SAMOCHÓD') then
 alter table Poz_Faktury
 delete foreign key FK_POZ_FAKT_SAMOCHOD__SAMOCHÓD
end if;*

*if exists(select 1 from sys.sysforeignkey where
role='FK_PRACOWNI_PRACOWNIK_SALONSAM') then
 alter table Pracownik
 delete foreign key FK_PRACOWNI_PRACOWNIK_SALONSAM
end if;*

*if exists(select 1 from sys.sysforeignkey where
role='FK_PRODUCEN_KRAJ_PROD_KRAJ') then
 alter table Producent
 delete foreign key FK_PRODUCEN_KRAJ_PROD_KRAJ
end if;*

*if exists(select 1 from sys.sysforeignkey where
role='FK_SAMOCHÓD_PRODUCENT_PRODUCEN') then
 alter table Samochód
 delete foreign key FK_SAMOCHÓD_PRODUCENT_PRODUCEN
end if;*

*if exists(select 1 from sys.sysforeignkey where
role='FK_SAMOCHÓD_SAMOCHODY_SALONSAM') then
 alter table Samochód
 delete foreign key FK_SAMOCHÓD_SAMOCHODY_SALONSAM
end if;*

drop index if exists Akcesoria.Akcesoria_w_salonie_FK;

drop index if exists Akcesoria.Typ_Akcesoriów_FK;

drop index if exists Akcesoria.Producent_Akcesoriów_FK;

drop index if exists Akcesoria.Akcesoria_PK;

drop table if exists Akcesoria;

drop index if exists Faktura.Wystawiajacy_Faktury_FK;

drop index if exists Faktura.Odbierajacy_faktury_FK;

drop index if exists Faktura.Faktura_PK;

drop table if exists Faktura;

drop index if exists Klient.Klient_PK;

drop table if exists Klient;

drop index if exists Kraj.Kraj_PK;

drop table if exists Kraj;

drop index if exists Poz_Faktury.Samochod_Pozycji_Faktury_FK;

drop index if exists Poz_Faktury.Pozycja_na_fakturze_FK;

drop index if exists Poz_Faktury.Akcesoria_pozycji_faktury_FK;

drop index if exists Poz_Faktury.Poz_Faktury_PK;

drop table if exists Poz_Faktury;

drop index if exists Pracownik.Pracownik_Salonu_FK;

drop index if exists Pracownik.Pracownik_PK;

drop table if exists Pracownik;

drop index if exists Producent.Kraj_Producenta_FK;

drop index if exists Producent.Producent_PK;

drop table if exists Producent;

drop index if exists SalonSamochodowy.SalonSamochodowy_PK;

drop table if exists SalonSamochodowy;

drop index if exists Samochód.Samochody_W_Salonie_FK;

drop index if exists Samochód.Producent_Pojazdu_FK;

drop index if exists Samochód.Samochód_PK;

drop table if exists Samochód;

drop index if exists Typ.Typ_PK;

drop table if exists Typ;

/=====**

/

/ Table: Akcesoria */*

/=====**

/

create table Akcesoria

(

IdSalonu integer not null,

IdAkcesorii integer not null,

```
IdTypu          integer          null,  
IdProducenta    integer          null,  
NazwaAkcesorii  varchar(50)      not null,  
CenaAkcesorii   numeric(15,2)    not null,  
constraint PK_AKCESORIA primary key (IdSalonu, IdAkcesorii)  
);
```

```
comment on table Akcesoria is  
'Akcesoria do samochodu np. Oleje, Wycieraczki, Koła  
';
```

```
comment on column Akcesoria.IdSalonu is  
'Identyfikator salonu';
```

```
comment on column Akcesoria.IdAkcesorii is  
'identyfikator akcesorii';
```

```
comment on column Akcesoria.IdTypu is  
'Identyfikator Typu  
';
```

```
comment on column Akcesoria.IdProducenta is  
'identyfikator producenta  
';
```

```
comment on column Akcesoria.NazwaAkcesorii is  
'nazwa akcesorii nadana przez producenta. ';
```

```
comment on column Akcesoria.CenaAkcesorii is  
'cena kupna akcesorii w salonie';
```

```
/*=====*/  
/  
/* Index: Akcesoria_PK */
```

```
/*=====*/  
/  
create unique index Akcesoria_PK on Akcesoria (  
  IdSalonu ASC,  
  IdAkcesorii ASC  
);
```

```
/*=====*/  
/  
/* Index: Producent_Akcesoriów_FK */  
/*=====*/  
/  
create index Producent_Akcesoriów_FK on Akcesoria (  
  IdProducenta ASC  
);
```

```
/*=====*/  
/  
/* Index: Typ_Akcesoriów_FK */  
/*=====*/  
/  
create index Typ_Akcesoriów_FK on Akcesoria (  
  IdTypu ASC  
);
```

```
/*=====*/  
/  
/* Index: Akcesoria_w_salonie_FK */  
/*=====*/  
/  
create index Akcesoria_w_salonie_FK on Akcesoria (  
  IdSalonu ASC  
);
```



```

/*=====
/
/* Table: Faktura                                */
/*=====
/
create table Faktura
(
    IdKlienta      integer      not null,
    IdFaktury       integer      not null,
    IdSalonu        integer      not null,
    IdPracownika    integer      not null,
    DataWystawienia date         null,
    constraint PK_FAKTURA primary key (IdKlienta, IdFaktury)
);

comment on table Faktura is
'jest to dokument zakupu towaru lub towarów w salonie';

comment on column Faktura.IdKlienta is
'identyfikator klienta';

comment on column Faktura.IdFaktury is
'identyfikator faktury';

comment on column Faktura.IdSalonu is
'Identyfikator salonu';

comment on column Faktura.IdPracownika is
'identyfikator pracownika';

comment on column Faktura.DataWystawienia is
'data wystawienia faktury';

/*=====
/

```

```

/* Index: Faktura_PK                                */
/*=====*
/
create unique index Faktura_PK on Faktura (
  IdKlienta ASC,
  IdFaktury ASC
);

/*=====*
/
/* Index: Odbierajacy_faktura_FK                    */
/*=====*
/
create index Odbierajacy_faktura_FK on Faktura (
  IdKlienta ASC
);

/*=====*
/
/* Index: Wystawiajacy_Faktura_FK                    */
/*=====*
/
create index Wystawiajacy_Faktura_FK on Faktura (
  IdSalonu ASC,
  IdPracownika ASC
);

/*=====*
/
/* Table: Klient                                    */
/*=====*
/
create table Klient
(
  IdKlienta      integer      not null,

```

```
Imie          varchar(30)          not null,  
Nazwisko      varchar(30)          not null,  
KodPocztowy   varchar(6)           null,  
Poczta        varchar(40)          null,  
Ulica         varchar(40)          null,  
NrDomu        varchar(10)          null,  
constraint PK_KLIENT primary key (IdKlienta)  
);
```

*comment on table Klient is
'jest to osoba kupująca towary w salonie';*

*comment on column Klient.IdKlienta is
'identyfikator klienta';*

*comment on column Klient.Imie is
'imie danej osoby';*

*comment on column Klient.Nazwisko is
'nazwisko danej osoby';*

*comment on column Klient.KodPocztowy is
'kod pocztowy miejsca zamieszkania danej osoby lub położenia salonu
samochodowego';*

*comment on column Klient.Poczta is
'nazwa miejscowości w której znajduje się poczta do której należy dana osoba';*

*comment on column Klient.Ulica is
'ulica na której znajduje się salon lub mieszka osoba';*

*comment on column Klient.NrDomu is
'numerr domu w którym mieszka dana osoba';*

```

/*=====*
/
/* Index: Klient_PK                               */
/*=====*
/
create unique index Klient_PK on Klient (
  IdKlienta ASC
);
/*=====*
/
/* Sequence: SEQ_Kraj                               */
/*=====*
/
CREATE SEQUENCE SEQ_Kraj START WITH 1 INCREMENT BY 1 MINVALUE 1 ;
/*=====*
/
/* Table: Kraj                                       */
/*=====*
/
create table Kraj
(
  IdKraju          integer DEFAULT (SEQ_Kraj.nextval) not null,
  NazwaKraju       varchar(30) not null,
  constraint PK_KRAJ primary key (IdKraju)
);

comment on table Kraj is
'Kraj pochodzenia producenta
';

comment on column Kraj.IdKraju is
'identyfikator kraju';

comment on column Kraj.NazwaKraju is

```

'Używana nazwa kraju';

```
/*=====*  
/  
/* Index: Kraj_PK */  
/*=====*  
/  
create unique index Kraj_PK on Kraj (  
IdKraju ASC  
);
```

```
/*=====*  
/  
/* Table: Poz_Faktury */  
/*=====*  
/  
create table Poz_Faktury  
(  
    IdKlienta      integer      not null,  
    IdFaktury      integer      not null,  
    IdPozFaktury   integer      not null,  
    IdSalonu       integer      null,  
    IdSamochodu    integer      null,  
    Akc_IdSalonu   integer      null,  
    IdAkcesorii    integer      null,  
    constraint PK_POZ_FAKTURY primary key (IdKlienta, IdFaktury, IdPozFaktury)  
);
```

comment on table Poz_Faktury is

*'jest to pozycja faktury oznaczajace jeden z akcesoriów lub samochodów
znajdujących sie na fakturze';*

comment on column Poz_Faktury.IdKlienta is

'identyfikator klienta';

*comment on column Poz_Faktury.IdFaktury is
'identyfikator faktury';*

*comment on column Poz_Faktury.IdPozFaktury is
'identyfikator pozycji faktury';*

*comment on column Poz_Faktury.IdSalonu is
'Identyfikator salonu';*

*comment on column Poz_Faktury.IdSamochodu is
'identyfikator samochodu';*

*comment on column Poz_Faktury.Akc_IdSalonu is
'Identyfikator salonu';*

*comment on column Poz_Faktury.IdAkcesorii is
'identyfikator akcesorii';*

```
/*=====*  
/  
/* Index: Poz_Faktury_PK */  
/*=====*  
/  
create unique index Poz_Faktury_PK on Poz_Faktury (  
IdKlienta ASC,  
IdFaktury ASC,  
IdPozFaktury ASC  
);
```

```
/*=====*  
/  
/* Index: Akcesoria_pozycji_faktury_FK */  
/*=====*  
/  
create index Akcesoria_pozycji_faktury_FK on Poz_Faktury (
```

```
Akc_IdSalonu ASC,  
IdAkcesorii ASC  
);
```

```
/*=====*/  
/  
/* Index: Pozycja_na_fakturze_FK */  
/*=====*/  
/  
create index Pozycja_na_fakturze_FK on Poz_Faktury (  
IdKlienta ASC,  
IdFaktury ASC  
);
```

```
/*=====*/  
/  
/* Index: Samochod_Pozycji_Faktury_FK */  
/*=====*/  
/  
create index Samochod_Pozycji_Faktury_FK on Poz_Faktury (  
IdSalonu ASC,  
IdSamochodu ASC  
);
```

```
/*=====*/  
/  
/* Table: Pracownik */  
/*=====*/  
/  
create table Pracownik  
(  
IdSalonu integer not null,  
IdPracownika integer not null,  
Imie varchar(30) not null,  
Nazwisko varchar(30) not null,
```

```
KodPocztowy    varchar(6)        null,  
Poczta         varchar(40)       null,  
Ulica          varchar(40)       null,  
NrDomu         varchar(10)       null,  
Stanowisko     varchar(30)       null,  
constraint PK_PRACOWNIK primary key (IdSalonu, IdPracownika)  
);
```

*comment on table Pracownik is
'jest to osoba pracująca w salonie';*

*comment on column Pracownik.IdSalonu is
'Identyfikator salonu';*

*comment on column Pracownik.IdPracownika is
'identyfikator pracownika';*

*comment on column Pracownik.Imie is
'imie danej osoby';*

*comment on column Pracownik.Nazwisko is
'nazwisko danej osoby';*

*comment on column Pracownik.KodPocztowy is
'kod pocztowy miejsca zamieszkania danej osoby lub położenia salonu
samochodowego';*

*comment on column Pracownik.Poczta is
'nazwa miejscowości w której znajduje się poczta do której należy dana osoba';*

*comment on column Pracownik.Ulica is
'ulica na której znajduje się salon lub mieszka osoba';*

*comment on column Pracownik.NrDomu is
'numerr domu w którym mieszka dana osoba';*

*comment on column Pracownik.Stanowisko is
'stanowisko zajmowane w salonie przez pracownika';*

```
/*=====*  
/  
/* Index: Pracownik_PK */  
/*=====*  
/  
create unique index Pracownik_PK on Pracownik (  
IdSalonu ASC,  
IdPracownika ASC  
);
```

```
/*=====*  
/  
/* Index: Pracownik_Salonu_FK */  
/*=====*  
/  
create index Pracownik_Salonu_FK on Pracownik (  
IdSalonu ASC  
);
```

```
/*=====*  
/  
/* Sequence: SEQ_Producent */
```

```
/*=====*  
/  
CREATE SEQUENCE SEQ_Producent START WITH 1 INCREMENT BY 1 MINVALUE  
1;  
/*=====*  
/  
/* Table: Producent */
```

```

/*=====*
/
create table Producent
(
    IdProducenta    integer DEFAULT (SEQ_Producent.nextval)    not null,
    IdKraju         integer                                not null,
    NazwaProducenta varchar(30)                                not null,
    constraint PK_PRODUCENT primary key (IdProducenta)
);

comment on table Producent is
'Producent sprzętu komputerowego';

comment on column Producent.IdProducenta is
'identyfikator producenta
';

comment on column Producent.IdKraju is
'identyfikator kraju';

comment on column Producent.NazwaProducenta is
'Potoczna nazwa producenta, np Lexus';

/*=====*
/
/* Index: Producent_PK */
/*=====*
/
create unique index Producent_PK on Producent (
    IdProducenta ASC
);

/*=====*
/
/* Index: Kraj_Producenta_FK */

```

```

/*=====*
/
create index Kraj_Producenta_FK on Producent (
IdKraju ASC
);

```

```

/*=====*
/
/* Table: SalonSamochodowy */
/*=====*
/
create table SalonSamochodowy
(
    IdSalonu      integer      not null,
    NazwaSalonu   varchar(30)   not null,
    Miejscowosc   varchar(30)   not null,
    KodPocztowy   varchar(6)    null,
    Ulica         varchar(40)    null,
    Ilosc_Modeli  Integer       null,
    constraint PK_SALONSAMOCODOWY primary key (IdSalonu)
);

```

comment on table SalonSamochodowy is
 'Salon Samochodowy jest to lokal, w ktorym klient może kupić samochód lub
 akcesoria do auta.';

comment on column SalonSamochodowy.IdSalonu is
 'Identyfikator salonu';

comment on column SalonSamochodowy.NazwaSalonu is
 'Jest to potoczna nazwa z jaka identyfikuje sie salon';

comment on column SalonSamochodowy.Miejscowosc is
 'Miejscowosc w jakiej znajduje sie salon';

*comment on column SalonSamochodowy.KodPocztowy is
'kod pocztowy miejsca zamieszkania danej osoby lub położenia salonu
samochodowego';*

*comment on column SalonSamochodowy.Ulica is
'ulica na ktorej znajduje sie salon lub mieszka osoba';*

*comment on column SalonSamochodowy.Ilosc_Modeli is
'Wartość określająca ile różnych modeli jest w danym salonie';*

```
/*=====*  
/  
/* Index: SalonSamochodowy_PK */  
/*=====*  
/  
create unique index SalonSamochodowy_PK on SalonSamochodowy (  
IdSalonu ASC  
);
```

```
/*=====*  
/  
/* Table: Samochód */  
/*=====*  
/  
create table Samochód  
(  
IdSalonu integer not null,  
IdSamochodu integer not null,  
IdProducenta integer null,  
Model varchar(50) null,  
Kolor varchar(15) null,  
RokProdukcji numeric(4) null,  
TypSilnika varchar(30) null,  
MocSilnika numeric(3) null,  
CenaSamochodu numeric(15,2) null,
```

constraint PK_SAMOCHÓD primary key (IdSalonu, IdSamochodu)
);

comment on table Samochód is
'Samochód dostępny do kupienia w salonie';

comment on column Samochód.IdSalonu is
'Identyfikator salonu';

comment on column Samochód.IdSamochodu is
'identyfikator samochodu';

comment on column Samochód.IdProducenta is
'identyfikator producenta
';

comment on column Samochód.Model is
'Model pojazdu danej marki';

comment on column Samochód.Kolor is
'Kolor samochodu';

comment on column Samochód.RokProdukcji is
'Rok w jakim dane auto zostało wyprodukowane';

comment on column Samochód.TypSilnika is
'Typ silnika mówiący nam o tym czym jest napędzany samochód';

comment on column Samochód.MocSilnika is
'Ilość koni mechanicznych';

comment on column Samochód.CenaSamochodu is
'Koszt zakupu danego pojazdu';

```

/*=====
/
/* Index: Samochód_PK */
/*=====
/
create unique index Samochód_PK on Samochód (
IdSalonu ASC,
IdSamochodu ASC
);

```

```

/*=====
/
/* Index: Producent_Pojazdu_FK */
/*=====
/
create index Producent_Pojazdu_FK on Samochód (
IdProducenta ASC
);

```

```

/*=====
/
/* Index: Samochody_W_Salonie_FK */
/*=====
/
create index Samochody_W_Salonie_FK on Samochód (
IdSalonu ASC
);

```

```

/*=====
/
/* Table: Typ */
/*=====
/
create table Typ
(

```

```
IdTypu          integer          not null,  
NazwaTypu       varchar(30)      not null,  
constraint PK_TYP primary key (IdTypu)  
);
```

comment on table Typ is

```
'Typ to dana kategoria akcesoriów sprzedawanych w salonie np. Wycieraczki,  
Olej  
';
```

comment on column Typ.IdTypu is

```
'Identyfikator Typu  
';
```

comment on column Typ.NazwaTypu is

```
'Potoczna nazwa danego typu towaru';
```

```
/*=====*/  
/  
/* Index: Typ_PK                               */  
/*=====*/  
/  
create unique index Typ_PK on Typ (  
IdTypu ASC  
);
```

alter table Akcesoria

```
add constraint FK_AKCESORI_AKCESORIA_SALONSAM foreign key (IdSalonu)  
references SalonSamochodowy (IdSalonu)  
on update restrict  
on delete restrict;
```

comment on foreign key Akcesoria.FK_AKCESORI_AKCESORIA_SALONSAM is

```
'Dzięki tej relacji określimy w jakim salonie znajdują się konkretne akcesoria';
```

alter table Akcesoria

*add constraint FK_AKCESORI_PRODUCENT_PRODUCEN foreign key
(IdProducenta)*

references Producent (IdProducenta)

on update restrict

on delete restrict;

*comment on foreign key Akcesoria.FK_AKCESORI_PRODUCENT_PRODUCEN is
'Dzięki tej relacji określimy kto produkuje dane akcesoria';*

alter table Akcesoria

add constraint FK_AKCESORI_TYP_AKCES_TYP foreign key (IdTypu)

references Typ (IdTypu)

on update restrict

on delete restrict;

*comment on foreign key Akcesoria.FK_AKCESORI_TYP_AKCES_TYP is
'Dzięki tej relacji możemy określić typ dla akcesoriów ';*

alter table Faktura

add constraint FK_FAKTURA_ODBIERAJA_KLIENT foreign key (IdKlienta)

references Klient (IdKlienta)

on update restrict

on delete restrict;

*comment on foreign key Faktura.FK_FAKTURA_ODBIERAJA_KLIENT is
'Dzięki tej relacji wiemy dla kogo została wystawiona faktura';*

alter table Faktura

*add constraint FK_FAKTURA_WYSTAWIAJ_PRACOWNI foreign key (IdSalonu,
IdPracownika)*

references Pracownik (IdSalonu, IdPracownika)

on update restrict

on delete restrict;

*comment on foreign key Faktura.FK_FAKTURA_WYSTAWIAJ_PRACOWNI is
'Dzięki tej relacji wiemy kto wystawił fakturę';*

alter table Poz_Faktury

*add constraint FK_POZ_FAKT_AKCESORIA_AKCESORI foreign key
(Akc_IdSalonu, IdAkcesorii)
references Akcesoria (IdSalonu, IdAkcesorii)
on update restrict
on delete restrict;*

*comment on foreign key Poz_Faktury.FK_POZ_FAKT_AKCESORIA_AKCESORI is
'Dzięki tej relacji wiemy jakie akcesoria i z jakiego salonu zajmują pozycję na
fakturze';*

alter table Poz_Faktury

*add constraint FK_POZ_FAKT_POZYCJA_N_FAKTURA foreign key (IdKlienta,
IdFaktury)
references Faktura (IdKlienta, IdFaktury)
on update restrict
on delete restrict;*

*comment on foreign key Poz_Faktury.FK_POZ_FAKT_POZYCJA_N_FAKTURA is
'Dzięki tej relacji wiemy jakie pozycje znajdują się na konkretnej fakturze';*

alter table Poz_Faktury

*add constraint FK_POZ_FAKT_SAMOCHOD__SAMOCHÓD foreign key
(IdSalonu, IdSamochodu)
references Samochód (IdSalonu, IdSamochodu)
on update restrict
on delete restrict;*

*comment on foreign key Poz_Faktury.FK_POZ_FAKT_SAMOCHOD__SAMOCHÓD
is
'Dzięki tej relacji wiemy jakie samochody i z jakiego salonu zajmują pozycję na
fakturze';*

alter table Pracownik

*add constraint FK_PRACOWNI_PRACOWNIK_SALONSAM foreign key
(IdSalonu)*

references SalonSamochodowy (IdSalonu)

on update restrict

on delete restrict;

alter table Producent

add constraint FK_PRODUCEN_KRAJ_PROD_KRAJ foreign key (IdKraju)

references Kraj (IdKraju)

on update restrict

on delete restrict;

*comment on foreign key Producent.FK_PRODUCEN_KRAJ_PROD_KRAJ is
'Dzięki tej relacji określimy w jakim kraju Producent produkuje swoje towary';*

alter table Samochód

*add constraint FK_SAMOCÓD_PRODUCENT_PRODUCEN foreign key
(IdProducenta)*

references Producent (IdProducenta)

on update restrict

on delete restrict;

*comment on foreign key Samochód.FK_SAMOCÓD_PRODUCENT_PRODUCEN
is*

'Dzięki tej relacji określimy kto produkuje dane Samochody';

alter table Samochód

*add constraint FK_SAMOCÓD_SAMOCODY_SALONSAM foreign key
(IdSalonu)*

references SalonSamochodowy (IdSalonu)

on update restrict

on delete restrict;

*comment on foreign key Samochód.FK_SAMochÓD_SAMochODY_SALONSAM
is*

*'Dzięki tej relacji określimy w jakim salonie znajdują się poszczególne
samochody';*

--KRAJ

*insert into Kraj (NazwaKraju) values ('Niemcy');
insert into Kraj (NazwaKraju) values ('Japonia');
insert into Kraj (NazwaKraju) values ('Francja');
insert into Kraj (NazwaKraju) values ('Wielka Brytania');
insert into Kraj (NazwaKraju) values ('USA');*

--Producenci Samochodow

*insert into Producent (IdKraju, NazwaProducenta) values (1, 'Mercedes');
insert into Producent (IdKraju, NazwaProducenta) values (1, 'Audi');
insert into Producent (IdKraju, NazwaProducenta) values (1, 'BMW');
insert into Producent (IdKraju, NazwaProducenta) values (2, 'Mazda');
insert into Producent (IdKraju, NazwaProducenta) values (2, 'Honda');
insert into Producent (IdKraju, NazwaProducenta) values (2, 'Mitsubishi');
insert into Producent (IdKraju, NazwaProducenta) values (3, 'Citroen');
insert into Producent (IdKraju, NazwaProducenta) values (3, 'Peugeot');
insert into Producent (IdKraju, NazwaProducenta) values (3, 'Renault');
insert into Producent (IdKraju, NazwaProducenta) values (4, 'Aston Martin');
insert into Producent (IdKraju, NazwaProducenta) values (4, 'Bentley');
insert into Producent (IdKraju, NazwaProducenta) values (4, 'Jaguar');
insert into Producent (IdKraju, NazwaProducenta) values (5, 'Ford');
insert into Producent (IdKraju, NazwaProducenta) values (5, 'Chevrolet');*

--Producenci Akcesoriow

*insert into Producent (IdProducenta, IdKraju, NazwaProducenta)
values
(15,1,'Bosch'),
(16,2,'Ashika'),
(17,4,'Garmin'),*

(18,2,'Bridgestone'),
(19,1,'Continental');

--SALONY

insert into SalonSamochodowy
(IdSalonu,NazwaSalonu,Miejscowosc,KodPocztowy,Ulica)
VALUES
(1,'Salon A','Warszawa','00-102','Marszałkowska'),
(2,'Salon B','Białystok','15-002','Sienkiewicza'),
(3,'Salon C','Łódź','90-004','Piotrkowska');

--TYPY AKCESORIÓW

INSERT into TYP (IdTypu,NazwaTypu)
VALUES
(1,'Wycieraczki'),
(2,'Opony'),
(3,'Nawigacja');

--AKCESORIA

insert into Akcesoria
(IdSalonu,IdAkcesorii,IdTypu,IdProducenta,NazwaAkcesorii,CenaAkcesorii)
VALUES
(1,1,1,15,'Wycieraczki BOSCH Aerotwin','110.95'),
(1,2,1,16,'Pióro Wycieraczki Ashika','35.00'),
(1,3,2,18,'Blizzak 205/55 R16 Zimowe','169.00'),
(1,4,2,18,'Blizzak 205/55 R16 Letnie','149.00'),
(1,5,3,17,'GARMIN Drive 50 LM Centralna Europa','499.99'),
(2,1,1,15,'Wycieraczki BOSCH Aerotwin','110.95'),
(2,2,1,16,'Pióro Wycieraczki Ashika','35.00'),
(2,3,2,18,'Blizzak 205/55 R16 Zimowe','169.00'),
(2,4,2,18,'Blizzak 205/55 R16 Letnie','149.00'),
(2,5,3,17,'GARMIN Drive 50 LM Centralna Europa','499.99'),
(3,1,1,15,'Wycieraczki BOSCH Aerotwin','110.95'),
(3,2,1,16,'Pióro Wycieraczki Ashika','35.00'),

(3,3,2,18,'WinterContact 155/70 R19 Zimowe','667.00'),
(3,4,2,18,'ContiEcoContact 205/55 R16 Letnie','286.00'),
(3,5,3,17,'GARMIN Drive 50 LM Centralna Europa','499.99');

--SAMOCHODY

INSERT into Samochod

(IdSalonu,IdSamochodu,IdProducenta,Model,Kolor,RokProdukcji,TypSilnika,Moc
Silnika,CenaSamochodu)

VALUES

(1,1,1,'Klasa S','srebrny','2017','Diesel','258','379000.00'),
(1,2,2,'Q5','Czarny','2017','Diesel','190','195000.00'),
(1,3,11,'Bentayga','srebrny','2016','Benzyna','608','729000.00'),
(1,4,13,'S-Max','czarny','2016','Benzyna','160','115900.00'),
(2,1,4,'6','niebieski','2016','Benzyna','165','112005.00'),
(2,2,6,'Outlander','brązowy metalik','2016','Diesel','150','150099.00'),
(3,1,12,'XF','czarny','2016','Benzyna','240','270920.00'),
(3,2,9,'Talisman','czerwony','2016','Benzyna','150','97082.00');

--KLIENCI

insert into Klient (IdKlienta, Imie, Nazwisko) values (1, 'Jan', 'Wiśniewski');
insert into Klient (IdKlienta, Imie, Nazwisko) values (2, 'Adam', 'Lewandowski');
insert into Klient (IdKlienta, Imie, Nazwisko) values (3, 'Damian', 'Zieliński');
insert into Klient (IdKlienta, Imie, Nazwisko) values (4, 'Urszula', 'Kozłowska');
insert into Klient (IdKlienta, Imie, Nazwisko) values (5, 'Michał', 'Krawczyk');
insert into Klient (IdKlienta, Imie, Nazwisko) values (6, 'Bogdan', 'Kaczmarek');
insert into Klient (IdKlienta, Imie, Nazwisko) values (7, 'Janusz', 'Piotrowski');
insert into Klient (IdKlienta, Imie, Nazwisko) values (8, 'Halina', 'Majewska');
insert into Klient (IdKlienta, Imie, Nazwisko) values (9, 'Henryk', 'Dudek');
insert into Klient (IdKlienta, Imie, Nazwisko) values (10, 'Mariusz', 'Dudek');

--PRACOWNICY

insert into Pracownik (IdPracownika, IdSalonu, Imie, Nazwisko) values (1, 1,
'Jan', 'Kowalski');
insert into Pracownik (IdPracownika, IdSalonu, Imie, Nazwisko) values (2, 1,
'Joanna', 'Nowak');

insert into Pracownik (IdPracownika, IdSalonu, Imie, Nazwisko) values (3, 1, 'Roman', 'Konarski');

insert into Pracownik (IdPracownika, IdSalonu, Imie, Nazwisko) values (1, 2, 'Marek', 'Piotrowski');

insert into Pracownik (IdPracownika, IdSalonu, Imie, Nazwisko) values (2, 2, 'Grzegorz', 'Brzydki');

insert into Pracownik (IdPracownika, IdSalonu, Imie, Nazwisko) values (1, 3, 'Mariusz', 'Abacki');

insert into Pracownik (IdPracownika, IdSalonu, Imie, Nazwisko) values (2, 3, 'Stanisław', 'Abacki');

insert into Pracownik (IdPracownika, IdSalonu, Imie, Nazwisko) values (3, 3, 'Jan', 'Pol');

--FAKTURY

insert into Faktura(IdKlienta, IdFaktury, IdPracownika, IdSalonu, DataWystawienia) values (1, 1, 1, 3, '2016-01-12');

insert into Faktura(IdKlienta, IdFaktury, IdPracownika, IdSalonu, DataWystawienia) values (2, 1, 2, 3, '2016-03-14');

insert into Faktura(IdKlienta, IdFaktury, IdPracownika, IdSalonu, DataWystawienia) values (3, 1, 2, 1, '2016-02-21');

insert into Faktura(IdKlienta, IdFaktury, IdPracownika, IdSalonu, DataWystawienia) values (3, 2, 2, 1, '2016-01-14');

insert into Faktura(IdKlienta, IdFaktury, IdPracownika, IdSalonu, DataWystawienia) values (4, 1, 2, 2, '2015-03-25');

insert into Faktura(IdKlienta, IdFaktury, IdPracownika, IdSalonu, DataWystawienia) values (5, 1, 2, 2, '2017-01-11');

```
insert into Faktura(IdKlienta, IdFaktury, IdPracownika,  
IdSalonu,DataWystawienia) values (6, 1, 1, 1,'2017-02-12');
```

```
insert into Faktura(IdKlienta, IdFaktury, IdPracownika,  
IdSalonu,DataWystawienia) values (7, 1, 2, 1,'2014-02-14');  
insert into Faktura(IdKlienta, IdFaktury, IdPracownika,  
IdSalonu,DataWystawienia) values (7, 2, 2, 1,'2015-12-09');
```

```
insert into Faktura(IdKlienta, IdFaktury, IdPracownika,  
IdSalonu,DataWystawienia) values (8, 1, 1, 3,'2014-06-05');
```

--Pozycje na Fakturach

//1

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdSamochodu,  
IdSalonu) values (1, 1, 1, 1, 3);  
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdAkcesorii,  
Akc_IdSalonu) values (1, 1, 2, 2, 3);
```

//2

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdSamochodu,  
IdSalonu) values (2, 1, 1, 2, 3);
```

//3

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdSamochodu,  
IdSalonu) values (3, 1, 1, 1, 1);  
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdAkcesorii,  
Akc_IdSalonu) values (3, 1, 2, 1, 1);
```

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdSamochodu, IdSalonu) values (3, 2, 1, 2, 1);
```

```
//4
```

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdSamochodu, IdSalonu) values (4, 1, 1, 2, 2);
```

```
//5
```

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdSamochodu, IdSalonu) values (5, 1, 1, 1, 2);
```

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdAkcesorii, Akc_IdSalonu) values (5, 1, 2, 1, 2);
```

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdAkcesorii, Akc_IdSalonu) values (5, 1, 3, 3, 2);
```

```
//6
```

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdSamochodu, IdSalonu) values (6, 1, 1, 2, 1);
```

```
//7
```

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdSamochodu, IdSalonu) values (7, 1, 1, 1, 2);
```

```
//8
```

```
insert into Poz_Faktury(IdKlienta, IdFaktury, IdPozFaktury, IdSamochodu, IdSalonu) values (8, 1, 1, 1, 3);
```

10. Wyniki wykonywania poszczególnych funkcji.

a) widok bestCarProducers

```
select * from bestCarProducers
```

	NazwaProducenta	NazwaKraju	LacznyPrzychod	LiczbaSprzedanych
1	Mercedes	Niemcy	379 000,00	1
2	Jaguar	Wielka Brytania	270 920,00	1
3	Audi	Niemcy	195 000,00	1
4	Mitsubishi	Japonia	150 099,00	1
5	Renault	Francja	97 082,00	1

b) widok bestCarCustomers

*select * from bestCarCustomers*

	Nazwisko	Imie	LiczbaKupionychS	KwotaKupionychS
1	Zieliński	Damina	2	574 000,00
2	Wiśniewski	Jan	1	270 920,00
3	Kozłowska	Urszula	1	150 099,00
4	Lewandowski	Adam	1	97 082,00

c)widok AkcesoriaBestSeller

*select * from AkcesoriaBestSeller*

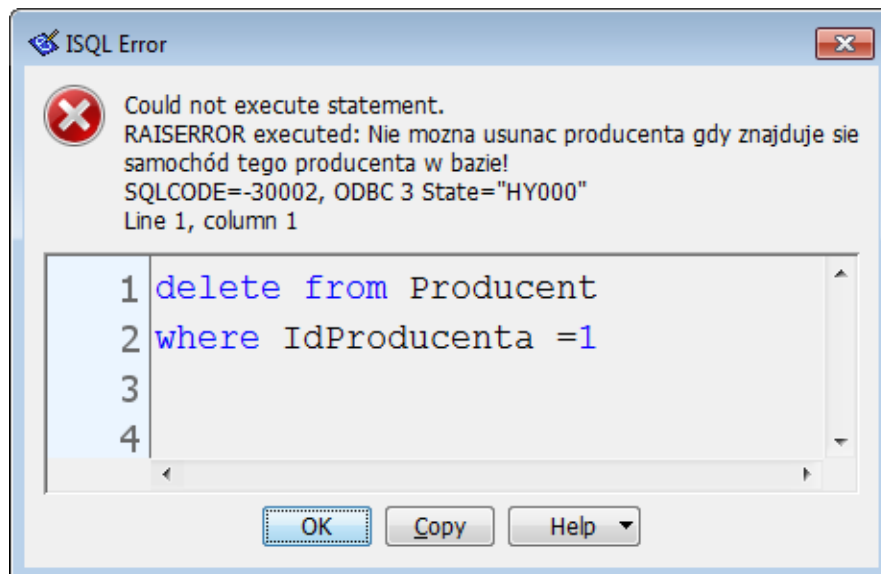
	NazwaTypu	NazwaAkcesorii	LiczbaSprzedanych	LacznyPrzychod
1	Wycieraczki	Piôro Wycieraczki Ashika	2	70,00
2	Wycieraczki	Wycieraczki BOSCH Aerotwin	1	110,95

d)widok AkcesoriaTotalValue

	IdSalonu	NazwaSalonu	Laczna_Wartosc
1	3	Salon C	1 598,94
2	1	Salon A	963,94
3	2	Salon B	963,94

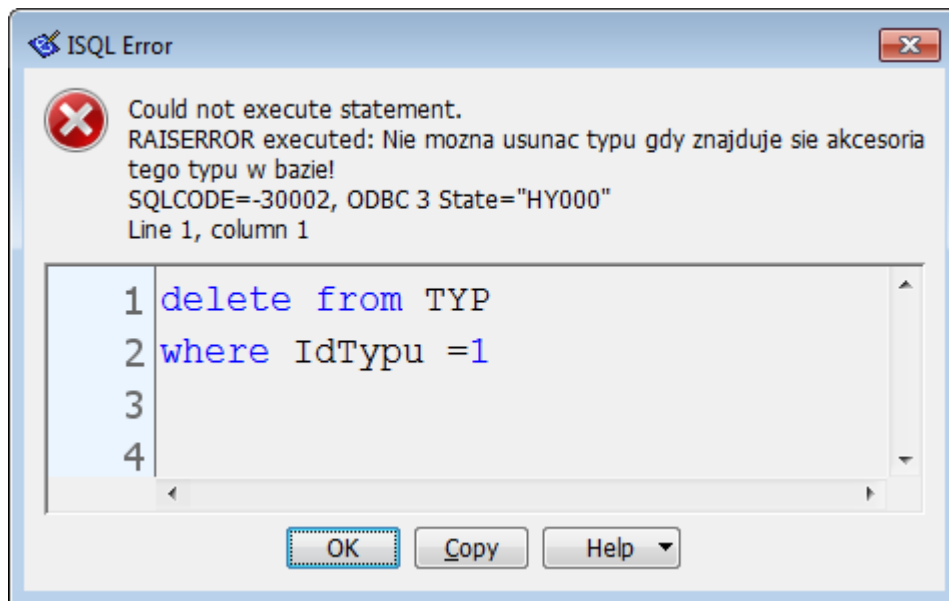
e)TRIG_Producent

delete from Producent where IdProducenta =1



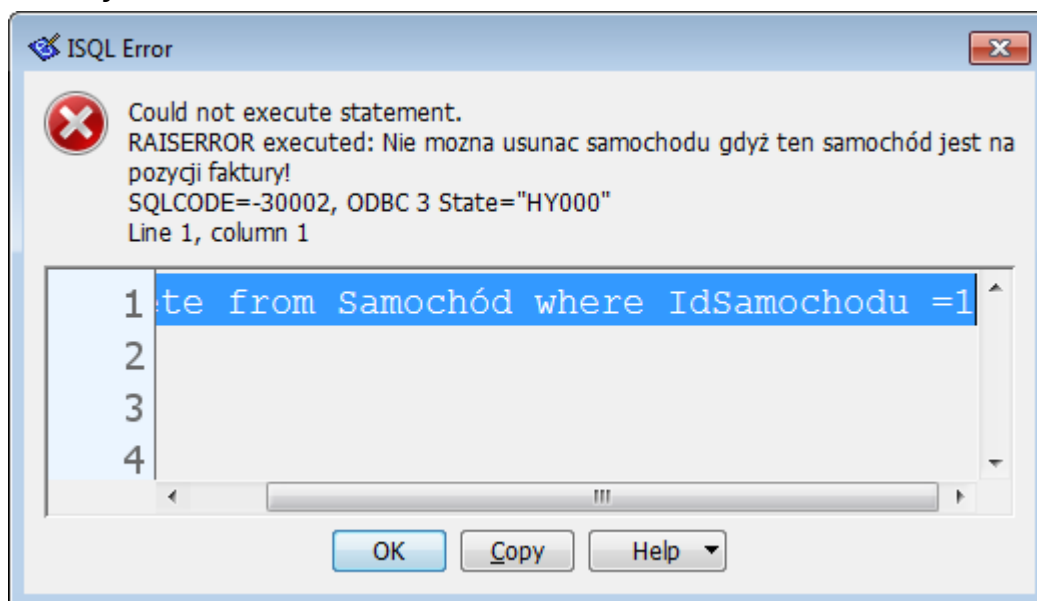
f)TRIG_Typ

delete from TYP where IdTypu =1



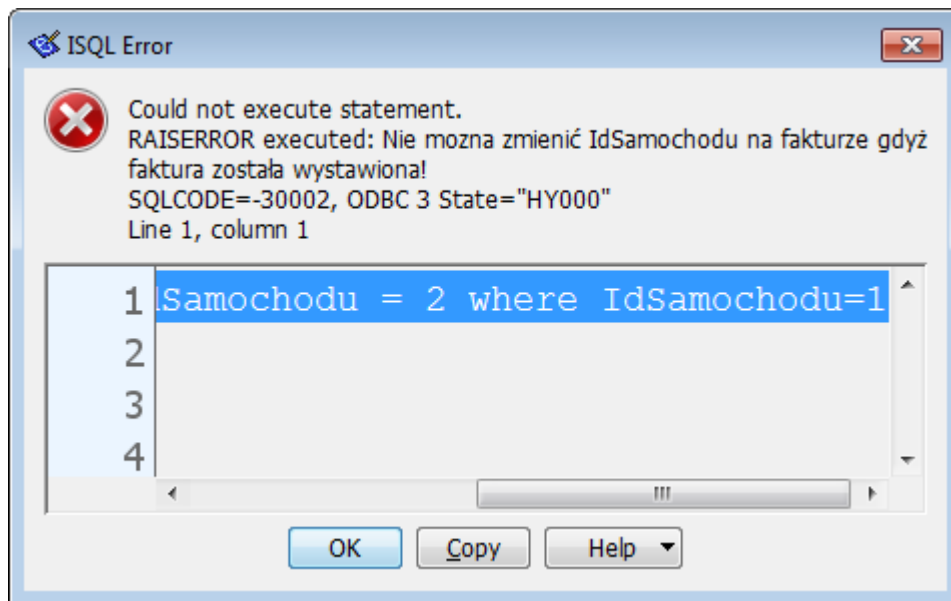
g)TRIG_Samochod

delete from Samochód where IdSamochodu =1



h)TRIG_PozFaktury

update Poz_Faktury Set IdSamochodu = 2 where IdSamochodu=1



i)TRIG_Salon

Przed Insertem

	IdSalonu	NazwaSalonu	Miejscowosc	KodPocztowy	Ulica	Ilosc_Modeli
1	1	Salon A	Warszawa	00-102	Marszałkowska	4
2	2	Salon B	Białystok	15-002	Sienkiewicza	2
3	3	Salon C	Łódź	90-004	Piotrkowska	2

Po Inscie nowego samochodu do salonu A

	IdSalonu	NazwaSalonu	Miejscowosc	KodPocztowy	Ulica	Ilosc_Modeli
1	1	Salon A	Warszawa	00-102	Marszałkowska	5
2	2	Salon B	Białystok	15-002	Sienkiewicza	2
3	3	Salon C	Łódź	90-004	Piotrkowska	2

i)Funkcja HowManyCars

select HowManyCars('mitsubishi')

	HowManyCars('mitsubishi')
1	1

j)Funkcja Kursor_New_Producent

select Kursor_New_Producent ('Opel','Niemcy')

	nowyP('Opel','Niemcy')
1	20

k)Funkcja Salon_Income

select Salon_Income(1)

	przychod(1)
1	574 145,95

l) Funkcja Pracownik_HowManyFactures

SELECT " Pracownik_HowManyFactures "("Mariusz','Abacki')"

DBA.wystawcy('Mariusz','Abacki')	
1	1

m) Procedura CarsSold

call CarsSold(1)

	IdSalonu	nazwa_salonu	marka_samochodu	model_samochodu	cena_samochodu
1	1	Salon A	Mercedes	Klasa S	379 000
2	1	Salon A	Audi	Q5	195 000

n) Procedura Kursor_New_Akcesoria

CALL " Kursor_New_Akcesoria "(" ProducentID " = 19," typID " = 2,"

AkcesoriaNazwa " = 'opony continental zimowe'," AkcesoriaCena " = 150)

Results

Towar dodany do bazy danych.
3 row(s) affected
Execution time: 0,12 seconds
Procedure completed

	IdSalonu	IdAkcesorii	IdTypu	IdProducenta	NazwaAkcesorii	CenaAkcesorii
1	1	1	1	15	Wycieraczki BOSCH Aerotwin	110,95
2	1	2	1	16	Pióro Wycieraczki Ashika	35,00
3	1	3	2	18	Blizzak 205/55 R16 Zimowe	169,00
4	1	4	2	18	Blizzak 205/55 R16 Letnie	149,00
5	1	5	3	17	Continental Drive 50 LR Centralna Europa	499,99
6	1	6	2	19	opony continental zimowe	150,00
7	2	1	1	15	Wycieraczki BOSCH Aerotwin	110,95
8	2	2	1	16	Pióro Wycieraczki Ashika	35,00
9	2	3	2	18	Blizzak 205/55 R16 Zimowe	169,00
10	2	4	2	18	Blizzak 205/55 R16 Letnie	149,00
11	2	5	3	17	Continental Drive 50 LR Centralna Europa	499,99
12	2	6	2	19	opony continental zimowe	150,00
13	3	1	1	15	Wycieraczki BOSCH Aerotwin	110,95
14	3	2	1	16	Pióro Wycieraczki Ashika	35,00
15	3	3	2	18	WinterContact 155/70 R19 Zimowe	667,00
16	3	4	2	18	ContiEcoContact 205/55 R16 Letnie	286,00
17	3	5	3	17	Continental Drive 50 LR Centralna Europa	499,99
18	3	6	2	19	opony continental zimowe	150,00

o) Procedura Klient_Purchased

CALL " Klient_Purchased "(" imieKlienta " = 'Damina'," nazwiskoKlienta " = 'Zieliński')"

	NazwaProducenta	NazwaAkcesorii	CenaAkcesorii
1	Bosch	Wycieraczki BOSCH Aerotwin	110,95
2	Ashika	PiÓro Wycieraczki Ashika	35,00
3	Mercedes	Klasa S	379 000,00
4	Audi	Q5	195 000,00

p) Procedura Salon_ShowAkcesoria

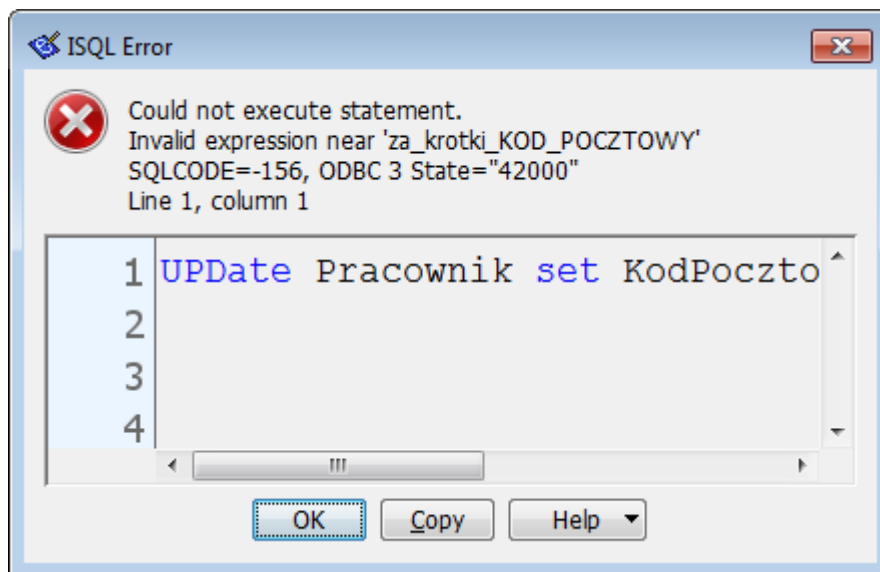
CALL "Salon_ShowAkcesoria "(" typTowaru " = 'SalonID ','@idS" = 1)

	NazwaAkcesorii	NazwaProducenta	CenaAkcesorii
1	Blizzak 205/55 ...	Bridgestone	169,00
2	Blizzak 205/55 ...	Bridgestone	149,00
3	opony continue...	Continental	150,00

q) TRIG_Pracownik

Po wpisaniu komendy

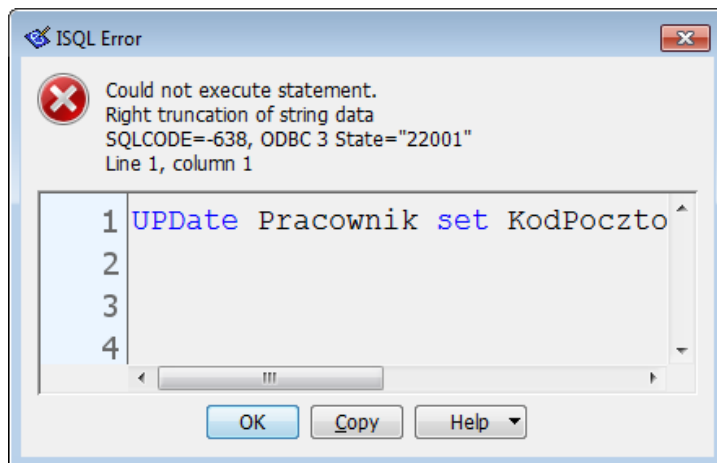
UPDATE Pracownik set KodPocztowy='17-0' where IdPracownika=1



Po wpisaniu komendy

UPDATE Pracownik set KodPocztowy='17-4220' where IdPracownika=1

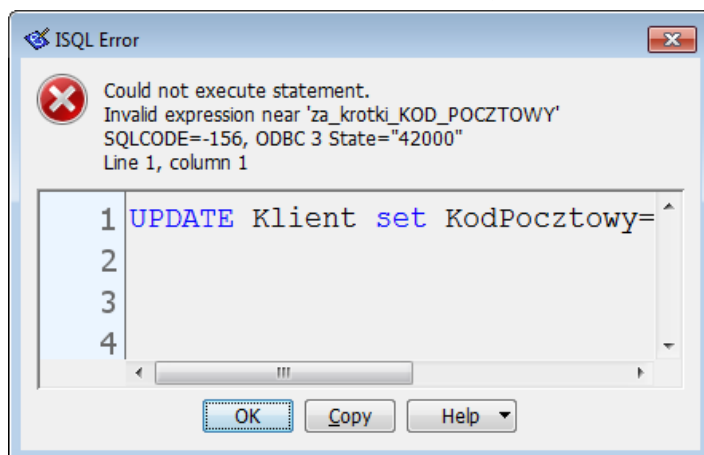
nie pojawił się komunikat wpisany w triggerze ponieważ przez to że ustawiliśmy w tabeli długość varchar dla kodu pocztowego varchar(6) pojawił się komunikat systemowy "Right truncation of string data"



r) TRIG_Klient

Po wpisaniu komendy

UPDATE Klient set KodPocztowy='17-0' where IdKlienta=1



Po wpisaniu komendy

UPDATE Klient set KodPocztowy='17-2340' where IdKlienta=1

