

WOJSKOWA AKADEMIA TECHNICZNA

PROJEKT LABORATORYJNY

KOMIS SAMOCHODOWY



Student: Więcek Patryk

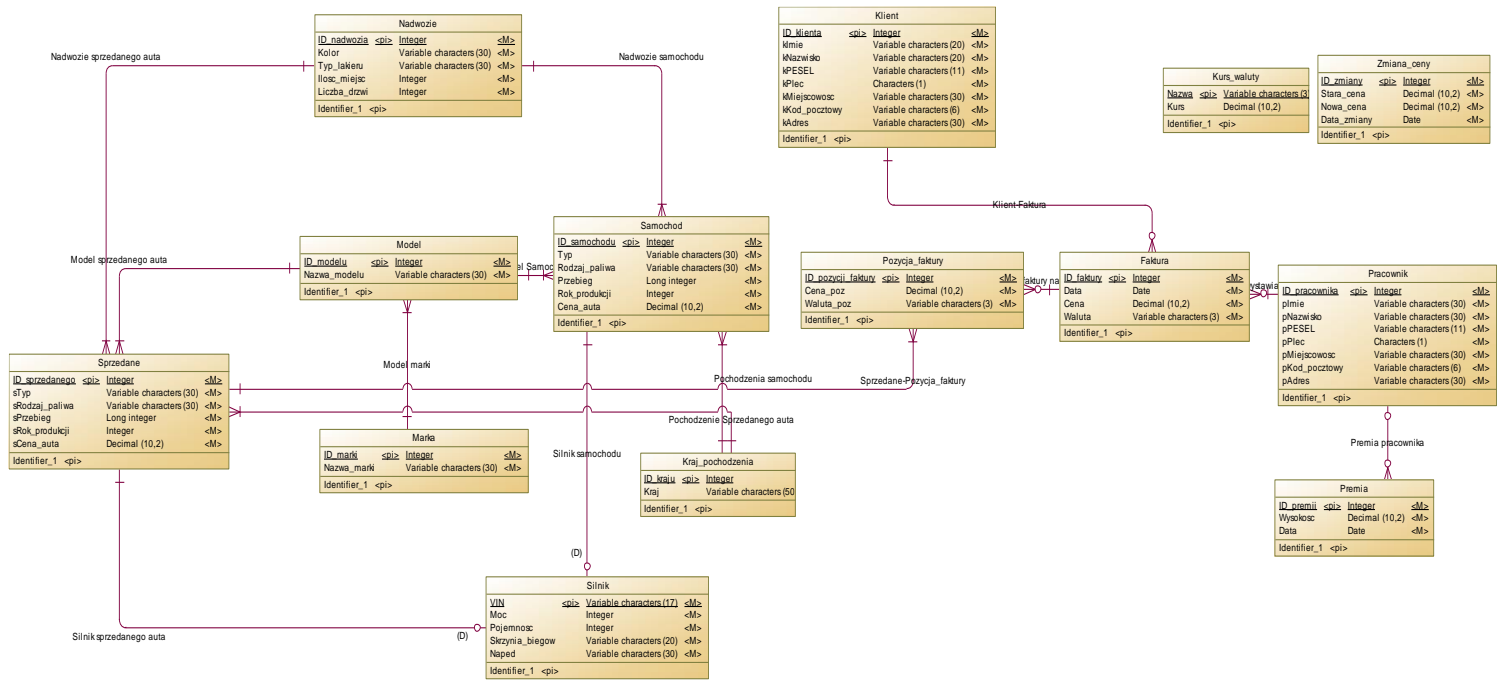
Grupa: WCY19IY2S1

Prowadzący: dr inż. Jarosław Koszela

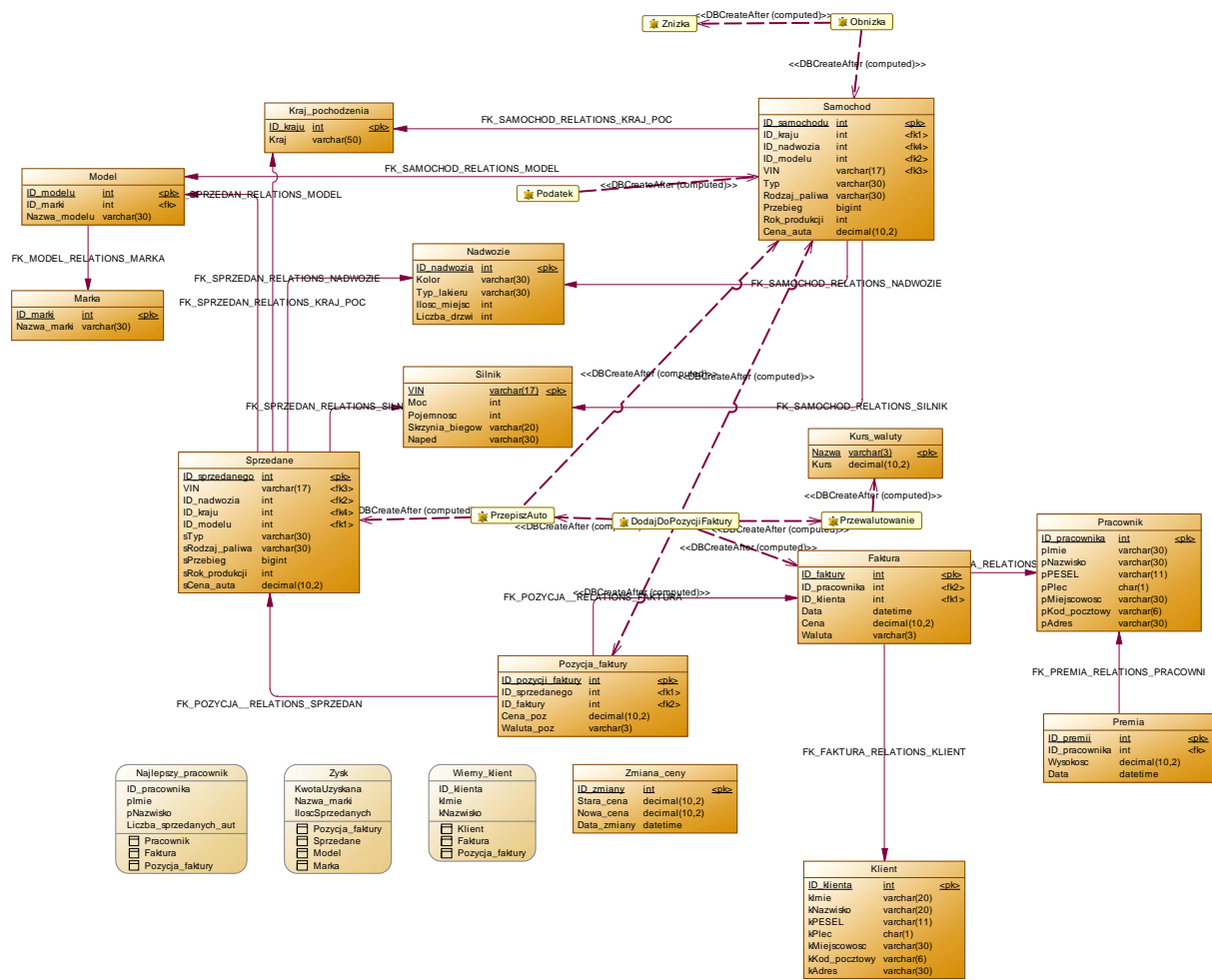
Analiza Biznesowa

Tematem bazy danych jest „Komis Samochodowy”. Zawarto w niej tabele z informacjami o cechach samochodu, kraju z którego pochodzą, specyfikacjami silników spalinowych w nich oraz historii ich sprzedaży. W bazie zawarte są również informacje o pracownikach w komisie, a także klientach, którzy dokonali zakupu w danym komisie. Owe tabele zawierają dane osobowe takie jak: imię, nazwisko, PESEL, płeć osoby, miejscowość, kod pocztowy oraz adres. Informacje o zakupie samochodów zawarte są w tabeli Pozycja_faktury, które później składają się na Fakturę, na której jest ID_faktury, Data wystawienia faktury, ID_klienta, czyli nowego właściciela, ID_pracownika, który wystawił fakturę, suma Cen z pozycji faktur oraz Waluta w jakiej została wystawiona. Natomiast dzięki tabeli Waluta, mamy dostęp do wybranych kursów walut, dzięki czemu można przeliczyć cenę auta ze złotych, na wybraną z tabeli walutę.

Konceptualny model bazy



Fizyczny model bazy



Kod generujący bazę danych

```
/*=====*/
/* DBMS name:   Microsoft SQL Server 2008          */
/* Created on:   2021-03-14 18:53:13                */
/*=====*/

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Faktura') and o.name = 'FK_FAKTURA_RELATIONS_KLIENT')
alter table Faktura
  drop constraint FK_FAKTURA_RELATIONS_KLIENT
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Faktura') and o.name = 'FK_FAKTURA_RELATIONS_PRACOWNI')
alter table Faktura
  drop constraint FK_FAKTURA_RELATIONS_PRACOWNI
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Model') and o.name = 'FK_MODEL_RELATIONS_MARKA')
alter table Model
  drop constraint FK_MODEL_RELATIONS_MARKA
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Pozycja_faktury') and o.name =
'FK_POZYCJA__RELATIONS_SPRZEDAN')
alter table Pozycja_faktury
  drop constraint FK_POZYCJA__RELATIONS_SPRZEDAN
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Pozycja_faktury') and o.name =
'FK_POZYCJA__RELATIONS_FAKTURA')
alter table Pozycja_faktury
  drop constraint FK_POZYCJA__RELATIONS_FAKTURA
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Premia') and o.name = 'FK_PREMIA_RELATIONS_PRACOWNI')
alter table Premia
  drop constraint FK_PREMIA_RELATIONS_PRACOWNI
go
```

```

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Samochod') and o.name = 'FK_SAMOCHOD_RELATIONS_KRAJ_POC')
alter table Samochod
  drop constraint FK_SAMOCHOD_RELATIONS_KRAJ_POC
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Samochod') and o.name = 'FK_SAMOCHOD_RELATIONS_MODEL')
alter table Samochod
  drop constraint FK_SAMOCHOD_RELATIONS_MODEL
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Samochod') and o.name = 'FK_SAMOCHOD_RELATIONS_SILNIK')
alter table Samochod
  drop constraint FK_SAMOCHOD_RELATIONS_SILNIK
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Samochod') and o.name = 'FK_SAMOCHOD_RELATIONS_NADWOZIE')
alter table Samochod
  drop constraint FK_SAMOCHOD_RELATIONS_NADWOZIE
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Sprzedane') and o.name = 'FK_SPRZEDAN_RELATIONS_MODEL')
alter table Sprzedane
  drop constraint FK_SPRZEDAN_RELATIONS_MODEL
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Sprzedane') and o.name = 'FK_SPRZEDAN_RELATIONS_NADWOZIE')
alter table Sprzedane
  drop constraint FK_SPRZEDAN_RELATIONS_NADWOZIE
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Sprzedane') and o.name = 'FK_SPRZEDAN_RELATIONS_SILNIK')
alter table Sprzedane
  drop constraint FK_SPRZEDAN_RELATIONS_SILNIK
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Sprzedane') and o.name = 'FK_SPRZEDAN_RELATIONS_KRAJ_POC')

```

```
alter table Sprzedane
  drop constraint FK_SPRZEDAN_RELATIONS_KRAJ_POC
go
```

```
if exists (select 1
  from sysindexes
  where id = object_id('Faktura')
  and name = 'Relationship_7_FK'
  and indid > 0
  and indid < 255)
  drop index Faktura.Relationship_7_FK
go
```

```
if exists (select 1
  from sysindexes
  where id = object_id('Faktura')
  and name = 'Relationship_5_FK'
  and indid > 0
  and indid < 255)
  drop index Faktura.Relationship_5_FK
go
```

```
if exists (select 1
  from sysobjects
  where id = object_id('Faktura')
  and type = 'U')
  drop table Faktura
go
```

```
if exists (select 1
  from sysobjects
  where id = object_id('Klient')
  and type = 'U')
  drop table Klient
go
```

```
if exists (select 1
  from sysobjects
  where id = object_id('Kraj_pochodzenia')
  and type = 'U')
  drop table Kraj_pochodzenia
go
```

```
if exists (select 1
  from sysobjects
  where id = object_id('Kurs_waluty')
  and type = 'U')
  drop table Kurs_waluty
go
```

```
if exists (select 1
  from sysobjects
```

```

        where id = object_id('Marka')
        and type = 'U')
    drop table Marka
go

if exists (select 1
    from sysindexes
    where id = object_id('Model')
    and name = 'Relationship_2_FK'
    and indid > 0
    and indid < 255)
    drop index Model.Relationship_2_FK
go

if exists (select 1
    from sysobjects
    where id = object_id('Model')
    and type = 'U')
    drop table Model
go

if exists (select 1
    from sysobjects
    where id = object_id('Nadwozie')
    and type = 'U')
    drop table Nadwozie
go

if exists (select 1
    from sysindexes
    where id = object_id('Pozycja_faktury')
    and name = 'Relationship_17_FK'
    and indid > 0
    and indid < 255)
    drop index Pozycja_faktury.Relationship_17_FK
go

if exists (select 1
    from sysindexes
    where id = object_id('Pozycja_faktury')
    and name = 'Relationship_9_FK'
    and indid > 0
    and indid < 255)
    drop index Pozycja_faktury.Relationship_9_FK
go

if exists (select 1
    from sysobjects
    where id = object_id('Pozycja_faktury')
    and type = 'U')
    drop table Pozycja_faktury
go

```



```
if exists (select 1
          from sysobjects
          where id = object_id('Pracownik')
          and type = 'U')
drop table Pracownik
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('Premia')
          and name = 'Relationship_18_FK'
          and indid > 0
          and indid < 255)
drop index Premia.Relationship_18_FK
go
```

```
if exists (select 1
          from sysobjects
          where id = object_id('Premia')
          and type = 'U')
drop table Premia
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('Samochod')
          and name = 'Relationship_11_FK'
          and indid > 0
          and indid < 255)
drop index Samochod.Relationship_11_FK
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('Samochod')
          and name = 'Relationship_10_FK'
          and indid > 0
          and indid < 255)
drop index Samochod.Relationship_10_FK
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('Samochod')
          and name = 'Relationship_8_FK'
          and indid > 0
          and indid < 255)
drop index Samochod.Relationship_8_FK
go
```

```
if exists (select 1
```

```

        from sysindexes
        where id = object_id('Samochochod')
        and name = 'Relationship_3_FK'
        and indid > 0
        and indid < 255)
    drop index Samochod.Relationship_3_FK
go

if exists (select 1
        from sysobjects
        where id = object_id('Samochochod')
        and type = 'U')
    drop table Samochod
go

if exists (select 1
        from sysobjects
        where id = object_id('Silnik')
        and type = 'U')
    drop table Silnik
go

if exists (select 1
        from sysindexes
        where id = object_id('Sprzedane')
        and name = 'Relationship_16_FK'
        and indid > 0
        and indid < 255)
    drop index Sprzedane.Relationship_16_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('Sprzedane')
        and name = 'Relationship_15_FK'
        and indid > 0
        and indid < 255)
    drop index Sprzedane.Relationship_15_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('Sprzedane')
        and name = 'Relationship_14_FK'
        and indid > 0
        and indid < 255)
    drop index Sprzedane.Relationship_14_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('Sprzedane')

```

```

        and name = 'Relationship_13_FK'
        and indid > 0
        and indid < 255)
drop index Sprzedane.Relationship_13_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('Sprzedane')
          and type = 'U')
drop table Sprzedane
go

if exists (select 1
          from sysobjects
          where id = object_id('Zmiana_ceny')
          and type = 'U')
drop table Zmiana_ceny
go

/*=====*/
/* Table: Faktura */
/*=====*/
create table Faktura (
  ID_faktury      int      not null,
  ID_pracownika   int      not null,
  ID_klienta      int      not null,
  Data           datetime not null,
  Cena           decimal(10,2) not null,
  Waluta         varchar(3) not null
  constraint CKC_WALUTA_FAKTURA check (Waluta >= '1' and Waluta = upper(Waluta)),
  constraint PK_FAKTURA primary key nonclustered (ID_faktury)
)
go

/*=====*/
/* Index: Relationship_5_FK */
/*=====*/
create index Relationship_5_FK on Faktura (
  ID_klienta ASC
)
go

/*=====*/
/* Index: Relationship_7_FK */
/*=====*/
create index Relationship_7_FK on Faktura (
  ID_pracownika ASC
)
go

/*=====*/

```

```

/* Table: Klient */
/*=====*/
create table Klient (
  ID_klienta      int          identity,
  klmie           varchar(20)   not null,
  kNazwisko       varchar(20)   not null,
  kPESEL          varchar(11)   not null
    constraint CKC_KPESEL_KLIENT check (kPESEL >= '11'),
  kPlec           char(1)       not null default 'M'
    constraint CKC_KPLEC_KLIENT check (kPlec = upper(kPlec)),
  kMiejscowosc    varchar(30)   not null,
  kKod_pocztowy   varchar(6)    not null,
  kAdres          varchar(30)   not null,
  constraint PK_KLIENT primary key nonclustered (ID_klienta)
)
go

/*=====*/
/* Table: Kraj_pochodzenia */
/*=====*/
create table Kraj_pochodzenia (
  ID_kraju        int          not null,
  Kraj            varchar(50)   not null,
  constraint PK_KRAJ_POCHODZENIA primary key nonclustered (ID_kraju)
)
go

/*=====*/
/* Table: Kurs_waluty */
/*=====*/
create table Kurs_waluty (
  Nazwa           varchar(3)    not null
    constraint CKC_NAZWA_KURS_WAL check (Nazwa in ('PLN','USD','EUR','RUR')),
  Kurs            decimal(10,2) not null,
  constraint PK_KURS_WALUTY primary key nonclustered (Nazwa)
)
go

/*=====*/
/* Table: Marka */
/*=====*/
create table Marka (
  ID_marki        int          not null,
  Nazwa_marki     varchar(30)   not null,
  constraint PK_MARKA primary key nonclustered (ID_marki)
)
go

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

```

```

ID_modelu      int          not null,
ID_marki       int          not null,
Nazwa_modelu   varchar(30)   not null,
constraint PK_MODEL primary key nonclustered (ID_modelu)
)
go

/*=====*/
/* Index: Relationship_2_FK                      */
/*=====*/
create index Relationship_2_FK on Model (
ID_marki ASC
)
go

/*=====*/
/* Table: Nadwozie                              */
/*=====*/
create table Nadwozie (
ID_nadwozia    int          not null,
Kolor          varchar(30)   not null,
Typ_lakieru    varchar(30)   not null,
Ilosc_miejsc   int          not null,
Liczba_drzwi   int          not null,
constraint PK_NADWOZIE primary key nonclustered (ID_nadwozia)
)
go

/*=====*/
/* Table: Pozycja_faktury                       */
/*=====*/
create table Pozycja_faktury (
ID_pozycji_faktury int      identity,
ID_sprzedanego    int      not null,
ID_faktury         int      not null,
Cena_poz          decimal(10,2) not null,
Waluta_poz        varchar(3)  not null,
constraint PK_POZYCJA_FAKTURY primary key nonclustered (ID_pozycji_faktury)
)
go

/*=====*/
/* Index: Relationship_9_FK                      */
/*=====*/
create index Relationship_9_FK on Pozycja_faktury (
ID_faktury ASC
)
go

/*=====*/
/* Index: Relationship_17_FK                     */
/*=====*/

```

```

create index Relationship_17_FK on Pozycja_faktury (
ID_sprzedanego ASC
)
go

/*=====*/
/* Table: Pracownik */
/*=====*/
create table Pracownik (
ID_pracownika int identity,
plmie varchar(30) not null,
pNazwisko varchar(30) not null,
pPESEL varchar(11) not null
constraint CKC_PPESEL_PRACOWNI check (pPESEL >= '11'),
pPlec char(1) not null default 'M'
constraint CKC_PPLEC_PRACOWNI check (pPlec = upper(pPlec)),
pMiejscowosc varchar(30) not null,
pKod_pocztowy varchar(6) not null,
pAdres varchar(30) not null,
constraint PK_PRACOWNIK primary key nonclustered (ID_pracownika)
)
go

/*=====*/
/* Table: Premia */
/*=====*/
create table Premia (
ID_premii int identity,
ID_pracownika int null,
Wysokosc decimal(10,2) not null,
Data datetime not null,
constraint PK_PREMIA primary key nonclustered (ID_premii)
)
go

/*=====*/
/* Index: Relationship_18_FK */
/*=====*/
create index Relationship_18_FK on Premia (
ID_pracownika ASC
)
go

/*=====*/
/* Table: Samochod */
/*=====*/
create table Samochod (
ID_samochodu int identity,
ID_kraju int not null,
ID_nadwozia int not null,
ID_modelu int not null,
VIN varchar(17) null

```

```

        constraint CKC_VIN_SAMOCHOD check (VIN is null or (VIN >= '17' and VIN = upper(VIN))),
Typ          varchar(30)      not null,
Rodzaj_paliwa  varchar(30)      not null,
Przebieg       bigint         not null,
Rok_produkcji  int            not null,
Cenaauta       decimal(10,2)   not null,
constraint PK_SAMOCHOD primary key nonclustered (ID_samochodu)
)
go

/*=====*/
/* Index: Relationship_3_FK */
/*=====*/
create index Relationship_3_FK on Samochod (
VIN ASC
)
go

/*=====*/
/* Index: Relationship_8_FK */
/*=====*/
create index Relationship_8_FK on Samochod (
ID_nadwozia ASC
)
go

/*=====*/
/* Index: Relationship_10_FK */
/*=====*/
create index Relationship_10_FK on Samochod (
ID_kraju ASC
)
go

/*=====*/
/* Index: Relationship_11_FK */
/*=====*/
create index Relationship_11_FK on Samochod (
ID_modelu ASC
)
go

/*=====*/
/* Table: Silnik */
/*=====*/
create table Silnik (
VIN          varchar(17)      not null
constraint CKC_VIN_SILNIK check (VIN >= '17' and VIN = upper(VIN)),
Moc          int              not null,
Pojemnosc    int              not null,
Skrzynia_biegow  varchar(20)   not null,
Naped        varchar(30)      not null,

```

```

    constraint PK_SILNIK primary key nonclustered (VIN)
)
go

/*=====*/
/* Table: Sprzedane */
/*=====*/
create table Sprzedane (
    ID_sprzedanego int not null,
    VIN varchar(17) null
    constraint CKC_VIN_SPRZEDAN check (VIN is null or (VIN >= '17' and VIN = upper(VIN))),
    ID_nadwozia int not null,
    ID_kraju int not null,
    ID_modelu int not null,
    sTyp varchar(30) not null,
    sRodzaj_paliwa varchar(30) not null,
    sPrzebieg bigint not null,
    sRok_produkcji int not null,
    sCena_auta decimal(10,2) not null,
    constraint PK_SPRZEDANE primary key nonclustered (ID_sprzedanego)
)
go

/*=====*/
/* Index: Relationship_13_FK */
/*=====*/
create index Relationship_13_FK on Sprzedane (
    ID_modelu ASC
)
go

/*=====*/
/* Index: Relationship_14_FK */
/*=====*/
create index Relationship_14_FK on Sprzedane (
    ID_nadwozia ASC
)
go

/*=====*/
/* Index: Relationship_15_FK */
/*=====*/
create index Relationship_15_FK on Sprzedane (
    VIN ASC
)
go

/*=====*/
/* Index: Relationship_16_FK */
/*=====*/
create index Relationship_16_FK on Sprzedane (
    ID_kraju ASC

```



```

)
go

/*=====*/
/* Table: Zmiana_ceny */
/*=====*/
create table Zmiana_ceny (
    ID_zmiany      int          identity,
    Stara_cena      decimal(10,2) not null,
    Nowa_cena       decimal(10,2) not null,
    Data_zmiany     datetime     not null,
    constraint PK_ZMIANA_CENY primary key nonclustered (ID_zmiany)
)
go

alter table Faktura
    add constraint FK_FAKTURA_RELATIONS_KLIENT foreign key (ID_klienta)
        references Klient (ID_klienta)
go

alter table Faktura
    add constraint FK_FAKTURA_RELATIONS_PRACOWNI foreign key (ID_pracownika)
        references Pracownik (ID_pracownika)
go

alter table Model
    add constraint FK_MODEL_RELATIONS_MARKA foreign key (ID_marki)
        references Marka (ID_marki)
go

alter table Pozycja_faktury
    add constraint FK_POZYCJA__RELATIONS_SPRZEDAN foreign key (ID_sprzedanego)
        references Sprzedane (ID_sprzedanego)
go

alter table Pozycja_faktury
    add constraint FK_POZYCJA__RELATIONS_FAKTURA foreign key (ID_faktury)
        references Faktura (ID_faktury)
go

alter table Premia
    add constraint FK_PREMIA_RELATIONS_PRACOWNI foreign key (ID_pracownika)
        references Pracownik (ID_pracownika)
go

alter table Samochod
    add constraint FK_SAMOCHOD_RELATIONS_KRAJ_POC foreign key (ID_kraju)
        references Kraj_pochodzenia (ID_kraju)
go

alter table Samochod
    add constraint FK_SAMOCHOD_RELATIONS_MODEL foreign key (ID_modelu)

```

```
references Model (ID_modelu)
go

alter table Samochod
add constraint FK_SAMOCOD_RELATIONS_SILNIK foreign key (VIN)
references Silnik (VIN)
go

alter table Samochod
add constraint FK_SAMOCOD_RELATIONS_NADWOZIE foreign key (ID_nadwozia)
references Nadwozie (ID_nadwozia)
go

alter table Sprzedane
add constraint FK_SPRZEDAN_RELATIONS_MODEL foreign key (ID_modelu)
references Model (ID_modelu)
go

alter table Sprzedane
add constraint FK_SPRZEDAN_RELATIONS_NADWOZIE foreign key (ID_nadwozia)
references Nadwozie (ID_nadwozia)
go

alter table Sprzedane
add constraint FK_SPRZEDAN_RELATIONS_SILNIK foreign key (VIN)
references Silnik (VIN)
go

alter table Sprzedane
add constraint FK_SPRZEDAN_RELATIONS_KRAJ_POC foreign key (ID_kraju)
references Kraj_pochodzenia (ID_kraju)
go
```

Procedury bazy danych

Dodaj do pozycji faktury

Procedura przyjmuje dane na temat ID_samochodu, ID_faktury, której będzie częścią, ID_pracownika, który wystawi fakturę, ID_klienta jako nowego właściciela samochodu oraz walutę w jakiej faktura zostanie wystawiona. Wykorzystywana jest tu funkcja Przewalutowanie, dzięki której ceny są przeliczane na wybraną z dostępnych walutę. Po dodaniu do faktury, auto jest przepisywane do tabeli Sprzedane i usuwane z tabeli Samochod, tak aby nie było problemów z określeniem, które są już niedostępne.

```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE OR
ALTER

PROCEDURE [dbo].[DodajDoPozycjiFaktury]
    @ID_samochodu INT
    ,@ID_faktury INT = NULL
    ,@ID_pracownika INT
    ,@ID_klienta INT
    ,@Wal VARCHAr (3)
AS
    IF EXISTS
        (
            SELECT 1
            FROM Samochod
            WHERE ID_samochodu = @ID_samochodu
        )
    BEGIN
    BEGIN TRY
        BEGIN TRAN

        IF NOT EXISTS (
            SELECT 1
            FROM Faktura
            WHERE ID_faktury = @ID_faktury
        )
        BEGIN
            INSERT INTO Faktura(
                ID_faktury
                ,ID_pracownika
                ,ID_klienta
                ,Cena
                ,Data
                ,Waluta
            )
            VALUES (
                @ID_faktury
                ,@ID_pracownika
                ,@ID_klienta
                ,0.0
                ,getdate()
                ,@Wal
            )
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRAN
    END CATCH
END
```

```

EXEC [dbo].[PrzepiszAuto] @ID_auta = @ID_samochodu
DECLARE @Cena DECIMAL(10,2)
SET @Cena =
    (
        SELECT Cena_auta
        FROM Samochod
        WHERE ID_samochodu = @ID_samochodu
    )

INSERT INTO Pozycja_faktury(
    ID_sprzedanego
    , ID_faktury
    , Cena_poz
    , Waluta_poz
)
VALUES (
    @ID_samochodu
    , @ID_faktury
    , ([dbo].[Przewalutowanie](@Cena, @Wal))
    , @Wal
)

UPDATE Faktura
SET Cena = (
    SELECT SUM(Cena_poz)
    FROM Pozycja_faktury
    WHERE ID_faktury = @ID_faktury
)
WHERE ID_faktury = @ID_faktury

IF @@TRANCOUNT > 0
    COMMIT TRAN
END TRY

BEGIN CATCH
    SELECT ERROR_NUMBER() AS ErrorNumber
    , ERROR_SEVERITY() AS ErrorSeverity
    , ERROR_STATE() AS ErrorState
    , ERROR_PROCEDURE() AS ErrorProcedure
    , ERROR_LINE() AS ErrorLine
    , ERROR_MESSAGE() AS ErrorMessage

    IF @@TRANCOUNT > 0
        ROLLBACK TRANSACTION;
END CATCH
END
ELSE
    BEGIN
        print 'Dany samochód nie jest w bazie'
    END

```

PrzepiszAuto

Procedura ma na celu przepisanie z tabeli Samochod auta, które zostało sprzedane. Przepisywane są wszystkie dane na podstawie otrzymanego ID_samochodu.

```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER

    PROCEDURE [dbo].[PrzepiszAuto]
        @ID_auta INT

AS
BEGIN TRY
    BEGIN TRAN
        DECLARE @ID_K INT
        DECLARE @ID_N INT
        DECLARE @ID_M INT
        DECLARE @sVIN VARCHAR(17)
        DECLARE @T VARCHAR(30)
        DECLARE @RP VARCHAR(30)
        DECLARE @P BIGINT
        DECLARE @R INT
        DECLARE @C DECIMAL(10,2)

        SET @ID_K = (
            SELECT ID_Kraju
            FROM Samochod
            WHERE ID_samochodu = @ID_auta
        )

        SET @ID_N = (
            SELECT ID_nadwozia
            FROM Samochod
            WHERE ID_samochodu = @ID_auta
        )

        SET @ID_M = (
            SELECT ID_modelu
            FROM Samochod
            WHERE ID_samochodu = @ID_auta
        )

        SET @sVIN = (
            SELECT VIN
            FROM Samochod
            WHERE ID_samochodu = @ID_auta
        )

        SET @T = (
            SELECT Typ
            FROM Samochod
            WHERE ID_samochodu = @ID_auta
        )

        SET @RP = (
            SELECT Rodzaj_paliwa
            FROM Samochod
            WHERE ID_samochodu = @ID_auta
        )

        SET @P = (
            SELECT Przebieg
            FROM Samochod
            WHERE ID_samochodu = @ID_auta
        )
    END TRY
END
```

```

        )
        SET @R = (
            SELECT Rok_produkcji
            FROM Samochod
            WHERE ID_samochodu = @ID_auta
        )

        SET @C = (
            SELECT Cena_auta
            FROM Samochod
            WHERE ID_samochodu = @ID_auta
        )

INSERT INTO Sprzedane
VALUES(@ID_auta, @sVIN, @ID_N, @ID_K, @ID_M, @T, @RP, @P, @R, @C)

IF @@TRANCOUNT > 0
    COMMIT TRAN
END TRY

BEGIN CATCH
    SELECT ERROR_NUMBER() AS ErrorNumber
    , ERROR_SEVERITY() AS ErrorSeverity
    , ERROR_STATE() AS ErrorState
    , ERROR_PROCEDURE() AS ErrorProcedure
    , ERROR_LINE() AS ErrorLine
    , ERROR_MESSAGE() AS ErrorMessage

    IF @@TRANCOUNT > 0
        ROLLBACK TRANSACTION;
END CATCH

```

Obnizka

Procedura obniża cenę samochodów z wybranego kraju o podany procent, podawany w postaci liczby po przecinku. Wykorzystuje ona funkcję Znizka.

```
USE [Proj v45]
GO
/***** Object:  StoredProcedure [dbo].[Obnizka]      Script Date: 2021-03-18 09:53:43
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER

    PROCEDURE [dbo].[Obnizka]
        @ID_k INT
        ,@Procent DECIMAL(4,2)

AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM Samochod
        WHERE ID_kraju = @ID_k
    )
    BEGIN
        DECLARE @Cena DECIMAL(10,2)
        DECLARE @ID_s INT
        DECLARE Coursorek CURSOR

        FOR
        SELECT ID_Samochodu, Cena_auta
        FROM Samochod
        WHERE ID_kraju = @ID_k

        OPEN Coursorek

        FETCH NEXT
        FROM Coursorek
        INTO @ID_s, @Cena

        IF @@FETCH_STATUS <> 0
            PRINT 'Pusty kursor'

        WHILE @@FETCH_STATUS = 0
        BEGIN
            UPDATE Samochod
            SET Cena_auta = dbo.Znizka(@Procent, @Cena)
            WHERE ID_samochodu = @ID_s

            FETCH NEXT
            FROM Coursorek
            INTO @ID_s, @Cena
        END

        CLOSE Coursorek

        DEALLOCATE Coursorek
    END --if1
END
```

Funkcje

Znizka

Funkcja po obniża cenę o podany w postaci dziesiętnej procent.

```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER

    FUNCTION [dbo].[Znizka] (

        @Procent AS DECIMAL(4,2)
        ,@Cena AS DECIMAL(10,2))
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @Promocja DECIMAL(10, 2)

    BEGIN
        SET @Promocja = 1 - @Procent
    END

    RETURN (cast(@Cena * @Promocja AS DECIMAL(10, 2)))
END
```


Przewalutowanie

Funkcja wykorzystuje tabelę Kurs_waluty w celu zmiany ceny na wybraną walutę z tabeli.

```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER

FUNCTION [dbo].[Przewalutowanie] (
    @Cena DECIMAL(10, 2)
    ,@Waluta VARCHAR (3)
)
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @Wynik DECIMAL(10,2)
    DECLARE @Kurs DECIMAL(10,2)
    SET @Kurs = (SELECT Kurs
                  FROM Kurs_waluty
                  WHERE Nazwa = @Waluta
                 )

    IF
        @Waluta = 'PLN'
    SET @Wynik = @Cena

    IF
        @Waluta = 'EUR'
    SET @Wynik = (cast(@Cena / @Kurs AS DECIMAL(10,2)))

    IF
        @Waluta = 'USD'
    SET @Wynik = (cast(@Cena / @Kurs AS DECIMAL(10,2)))

    IF
        @Waluta = 'RUR'
    SET @Wynik = (cast(@Cena / @Kurs AS DECIMAL(10,2)))

    RETURN @Wynik
END
```

Podatek

Funkcja wylicza jaki podatek trzeba będzie odprowadzić do urzędu skarbowego po zakupie auta.

```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER

FUNCTION [dbo].[Podatek] (
    @Podatek AS DECIMAL(4, 2)
    ,@ID_S AS DECIMAL(10,2)
)
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @Cena_auta DECIMAL(10,2)

    SET @Cena_auta = (
        SELECT Cena_auta
        FROM Samochod
        WHERE ID_samochodu = @Id_S
    )

    RETURN (cast(@Cena_auta * @Podatek AS DECIMAL(10, 2)))
END
```

Triggery

SprzedaneAuto

Po dodaniu samochodu do Pozycji_faktury zostaje on usunięty z tabeli Samochod.

```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER

    TRIGGER [dbo].[SprzedaneAuto] ON [dbo].[Pozycja_faktury]
AFTER INSERT
AS
BEGIN
    DECLARE @ID_Poz INT
    DECLARE @ID_S INT

    SET @ID_Poz = (
        SELECT MAX(ID_Pozycji_faktury)
        FROM Pozycja_faktury
    )

    SET @ID_S = (
        SELECT ID_sprzedanego
        FROM Pozycja_faktury
        WHERE ID_pozycji_faktury = @ID_Poz
    )

    DELETE FROM Samochod
    WHERE ID_samochodu = @ID_S
END
```

Premka

Na podstawie łącznej ceny sprzedanych aut, wyliczana jest roczna premia w wysokości 1% od tej wartości. Wysokość premii można zobaczyć w tabeli Premia.

```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER

    TRIGGER [dbo].[Premka] ON [dbo].[Faktura]
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @ID_F INT
    DECLARE @ID_P INT
    DECLARE @Cena DECIMAL(10,2)
    DECLARE @Wys DECIMAL(10,2)
    DECLARE @D DATETIME
    DECLARE @Wal AS VARCHAR(3)
    DECLARE @Kurs AS DECIMAL(10,2)
```

```

SET @ID_F = (
    SELECT ID_faktury
    FROM inserted
)

SET @ID_P = (
    SELECT ID_pracownika
    FROM Faktura
    WHERE ID_faktury = @ID_F
)

SET @Wal = (
    SELECT Waluta
    FROM Faktura
    WHERE ID_faktury = @ID_F
)

SET @Cena = (
    SELECT Cena
    FROM Faktura
    WHERE ID_faktury = @ID_F
)

SET @Kurs = (
    SELECT Kurs
    FROM Kurs_waluty
    WHERE Nazwa = @Wal
)

IF @Wal = 'PLN'
    BEGIN
        SET @Wys = (cast(@Cena * 1 * 0.01 AS DECIMAL(10,2)))
    END
ELSE
    BEGIN
        SET @Wys = (cast(@Cena * @Kurs * 0.01 AS DECIMAL(10,2)))
    END

SET @D = getdate()

IF NOT EXISTS (
    SELECT 1
    FROM Premia
    WHERE ID_pracownika = @ID_P AND year([Data]) =
year(@D)
)
    BEGIN
        INSERT INTO Premia
        VALUES(@ID_P, @Wys, @D)
    END
ELSE
    BEGIN
        UPDATE Premia
        SET Wysokosc += @Wys
        WHERE ID_pracownika = @ID_P AND year([Data]) = year(@D)
    END
END

```

ZmianaCeny

Trigger ten zapisuje zmiany cen, jakie zostały wprowadzone na Fakturze, aby zabezpieczyć niezgodność sumy wartości z pozycji faktury a samą fakturą. Zmiany zapisywane są w tabeli Zmiana_ceny.

```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER

TRIGGER [dbo].[ZmianaCeny] ON [dbo].[Faktura]
INSTEAD OF UPDATE
AS
BEGIN
    DECLARE @ID_F INT
    DECLARE @Cena DECIMAL(10,2)
    DECLARE @NCena DECIMAL(10,2)

    SET @ID_F = (
        SELECT ID_faktury
        FROM inserted
    )

    SET @Cena = (
        SELECT Cena
        FROM Faktura
        WHERE ID_faktury = @ID_F
    )

    SET @NCena =(
        SELECT Cena
        FROM inserted
    )

    INSERT INTO Zmiana_ceny (Stara_cena, Nowa_cena, Data_zmiany)
    VALUES (@Cena, @NCena, getdate())
    UPDATE Faktura
    SET [Data] = (
        SELECT [Data]
        FROM inserted
    ),

    Cena = (
        SELECT Cena
        FROM inserted
    ),

    Waluta = (
        SELECT Waluta
        FROM inserted
    )
    WHERE ID_faktury = @ID_F

END
```

Widoki

Najlepszy pracownik

Pokazuje pracownika, który sprzedał najwięcej samochodów w komisie.

```
CREATE VIEW Najlepszy_pracownik AS
SELECT p.ID_pracownika, p.pImie AS Imie_pracownika, p.pNazwisko AS
Nazwisko_pracownika, count(po.ID_pozycji_faktury) AS Liczba_sprzedanych_aut
FROM Pracownik p
    inner join Faktura f on p.ID_pracownika = f.ID_pracownika
    inner join Pozycja_faktury po on f.ID_faktury = po.ID_faktury
GROUP BY p.ID_pracownika, p.pImie, p.pNazwisko
HAVING count(po.ID_pozycji_faktury) =(
    SELECT top 1 count(fak.ID_pracownika)
    FROM Pozycja_faktury poz
        join Faktura fak on poz.ID_faktury = fak.ID_faktury
    WHERE month(fak.[Data]) = month(getdate())
    ORDER BY 1 desc
)
```

Wierny klient

Pokazuje klienta, który zakupił najwięcej samochodów w komisie.

```
CREATE VIEW Wierny_klient AS
SELECT k.ID_klienta, k.kImie AS Imie_klienta, k.kNazwisko AS Nazwisko_klienta
FROM Klient k
    inner join Faktura f on k.ID_klienta = f.ID_klienta
    inner join Pozycja_faktury po on f.ID_faktury = po.ID_faktury
GROUP BY k.ID_klienta, k.kImie, k.kNazwisko
HAVING count(po.ID_pozycji_faktury) =(
    SELECT top 1 count(fak.ID_klienta)
    FROM Pozycja_faktury poz
        join Faktura fak on poz.ID_faktury = fak.ID_faktury
    GROUP BY ID_klienta
    ORDER BY 1 desc
)
```

Zysk

Pokazuje ilość samochodów sprzedanych z danej marki oraz jaki zysk z nich był.

```
CREATE VIEW Zysk AS
SELECT sum(sp.sCena_auta) AS KwotaUzyskana, mar.Nazwa_marki,
count(sp.ID_sprzedanego) AS IloscSprzedanych
FROM Pozycja_faktury pf
    inner join Sprzedane sp on sp.ID_sprzedanego = pf.ID_sprzedanego
    inner join Model m on m.ID_modelu = sp.ID_modelu
    inner join Marka mar on mar.ID_marki = m.ID_marki
GROUP BY mar.Nazwa_marki
```

Użytkownicy

Utworzenie użytkownika Administrator, który jest właścicielem bazy, Pracownik z możliwością pisania oraz odczytywania tabel oraz Viewer, który jest klientem i może tylko przeglądać tabele.

```
CREATE LOGIN Administrator WITH PASSWORD = 'admin';  
go
```

```
CREATE USER [ADMINISTRATOR] FOR LOGIN Administrator
```

```
EXEC sp_addrolemember 'db_owner', 'ADMINISTRATOR'  
go
```

```
-----  
  
CREATE LOGIN Pracownik WITH PASSWORD = 'work';  
go
```

```
CREATE USER PRACOWNIK FOR LOGIN Pracownik
```

```
EXEC sp_addrolemember 'db_datareader', 'PRACOWNIK'  
EXEC sp_addrolemember 'db_datawriter', 'PRACOWNIK'  
go
```

```
-----  
  
CREATE LOGIN Viewer WITH PASSWORD = 'view';  
go
```

```
CREATE USER [VIEWER] FOR LOGIN Viewer
```

```
EXEC sp_addrolemember 'db_datareader', 'VIEWER'  
go
```