

23.03.2019

WOJSKOWA AKADEMIA TECHNICZNA

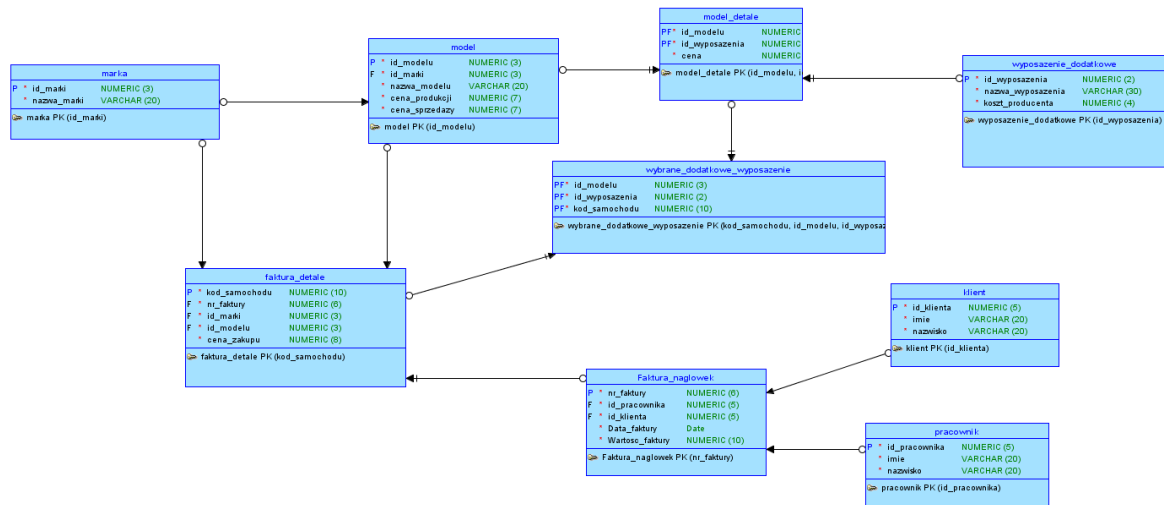
***BAZY DANYCH PROJEKT***

Student: Patryk Wiśniewski

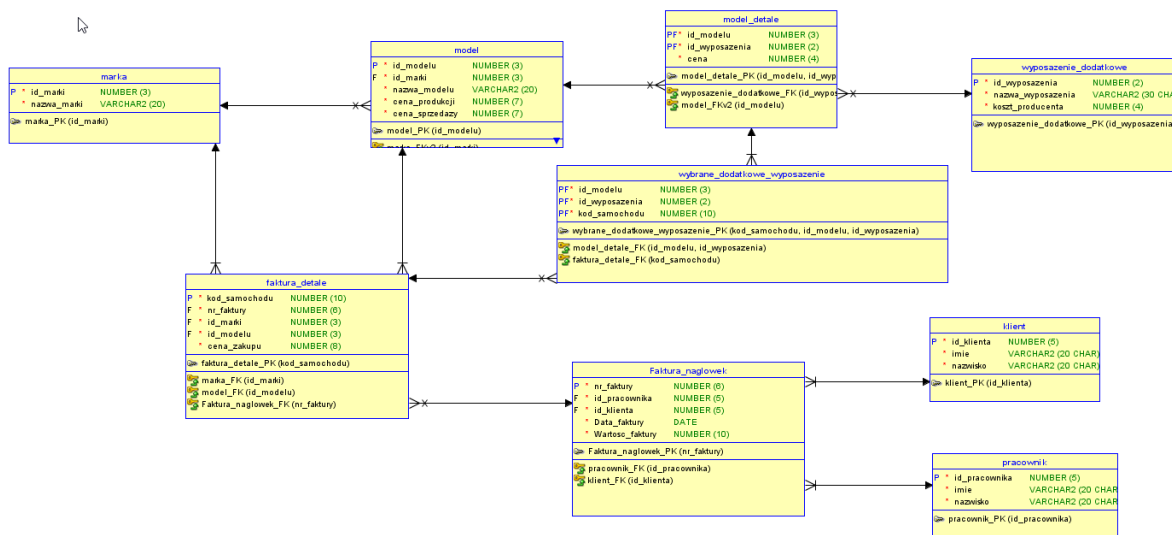
Grupa: I7Y2S1

Prowadzący: Józef Woźniak

## Model logiczny:



## Model relacyjny:



Projekt obrazowuje dzialanie pewnego Salonu Samochodowego. Obiekty w tym salonie nie są dostępne od ręki. Klient ma możliwość zapoznania się z oferowanymi modelami poprzez oglądanie modeli które nie sa przeznaczone na sprzedaż. W momencie podjęcia decyzji o zakupie może dokonać zamówienia. Na jednej fakturze może znajdować się wiele aut. Każde z nich reprezentowane jest poprzez unikatowy numer nadawany podczas składania zamówienia. W razie pomyłki wprowadzania danych do faktury wystarczy usunąć jej nagłówek, reszta zwalnia się automatycznie. Wprowadzanie nowych zamówień do bazy danych należy zaczynać kolejno od wprowadzania ich do tabel:

- 1.Klient
- 2.Faktura\_nagłówek.
- 3.Faktura\_detale.
- 4.wybrane\_dodatkowe\_wyposazenie

```

--tworzenie sekwencji uzywanych w projekcie
CREATE SEQUENCE seq_faktura_detale START WITH 1 INCREMENT BY 3 MAXVALUE 9999999999999999 MINVALUE 1 NOCACHE;
CREATE SEQUENCE seq_faktura_naglowek START WITH 1 INCREMENT BY 1 MAXVALUE 99999999999999999999 MINVALUE 1 NOCACHE;
CREATE SEQUENCE SEQ_KLIENT INCREMENT BY 1 MAXVALUE 9999999999 MINVALUE 1 NOCACHE;
CREATE SEQUENCE SEQ_MODEL INCREMENT BY 1 MAXVALUE 9999999999 MINVALUE 1 CACHE 20;
CREATE SEQUENCE SEQ_PRACOWNIK INCREMENT BY 1 MAXVALUE 99999999999999999999 MINVALUE 1 NOCACHE;
CREATE SEQUENCE SEQ_WYP_DODAT INCREMENT BY 1 MAXVALUE 99999999999999999999 MINVALUE 1 NOCACHE;
CREATE SEQUENCE SEQ_WYP_DODAT_KOSZT INCREMENT BY 1000 MAXVALUE 99999999999999999999 MINVALUE 0 NOCACHE;

--tworzenie tabel oraz dodawanie do nich kluczy
CREATE TABLE faktura_detale (
    kod_samochodu    NUMBER(10) NOT NULL,
    nr_faktury       NUMBER(6) NOT NULL,
    id_marki         NUMBER(3) NOT NULL,
    id_modelu        NUMBER(3) NOT NULL,
    cena_zakupu      NUMBER(8) NOT NULL
)
LOGGING;

ALTER TABLE faktura_detale ADD CONSTRAINT faktura_detale_pk PRIMARY KEY ( kod_samochodu );

CREATE TABLE faktura_naglowek (
    nr_faktury       NUMBER(6) NOT NULL,
    id_pracownika    NUMBER(5) NOT NULL,
    id_klienta       NUMBER(5) NOT NULL,
    data_faktury     DATE NOT NULL,
    wartosc_faktury  NUMBER(10) NOT NULL
)
LOGGING;

ALTER TABLE faktura_naglowek ADD CONSTRAINT faktura_naglowek_pk PRIMARY KEY ( nr_faktury );

CREATE TABLE klient (
    id_klienta       NUMBER(5) NOT NULL,
    imie             VARCHAR2(20 CHAR) NOT NULL,
    nazwisko         VARCHAR2(20 CHAR) NOT NULL
)
LOGGING;

ALTER TABLE klient ADD CONSTRAINT klient_pk PRIMARY KEY ( id_klienta );

CREATE TABLE marka (
    id_marki         NUMBER(3) NOT NULL,
    nazwa_marki      VARCHAR2(20) NOT NULL
)
LOGGING;

ALTER TABLE marka ADD CONSTRAINT marka_pk PRIMARY KEY ( id_marki );

CREATE TABLE model (
    id_modelu        NUMBER(3) NOT NULL,
    id_marki         NUMBER(3) NOT NULL,
    nazwa_modelu     VARCHAR2(20) NOT NULL,
    cena_produkcji   NUMBER(7) NOT NULL,
    cena_sprzedazy   NUMBER(7) NOT NULL
)
LOGGING;

ALTER TABLE model ADD CONSTRAINT model_pk PRIMARY KEY ( id_modelu );

CREATE TABLE model_detale (
    id_modelu        NUMBER(3) NOT NULL,
    id_wyposazenia   NUMBER(2) NOT NULL,
    cena             NUMBER(5) NOT NULL
)
LOGGING;

ALTER TABLE model_detale ADD CONSTRAINT model_detale_pk PRIMARY KEY ( id_modelu,
                                                                    id_wyposazenia );

CREATE TABLE pracownik (
    id_pracownika    NUMBER(5) NOT NULL,
    imie             VARCHAR2(20 CHAR) NOT NULL,
    nazwisko         VARCHAR2(20 CHAR) NOT NULL
)
LOGGING;

ALTER TABLE pracownik ADD CONSTRAINT pracownik_pk PRIMARY KEY ( id_pracownika );

CREATE TABLE wybrane_dodatkowe_wyposazenie (
    id_modelu        NUMBER(3) NOT NULL,
    id_wyposazenia   NUMBER(2) NOT NULL,
    kod_samochodu    NUMBER(10) NOT NULL
)
LOGGING;

```

```

-- ERROR: PK name length exceeds maximum allowed length(30)
ALTER TABLE wybrane_dodatkowe_wyposazenie
ADD CONSTRAINT wybrane_dodatkowe_wyposazenie_pk PRIMARY KEY ( kod_samochodu,
                                                                id_modelu,
                                                                id_wyposazenia );

CREATE TABLE wyposazenie_dodatkowe (
    id_wyposazenia    NUMBER(2) NOT NULL,
    nazwa_wyposazenia VARCHAR2(30 CHAR) NOT NULL,
    koszt_producenta  NUMBER(5) NOT NULL
)
LOGGING;

ALTER TABLE wyposazenie_dodatkowe ADD CONSTRAINT wyposazenie_dodatkowe_pk PRIMARY KEY ( id_wyposazenia );

ALTER TABLE wybrane_dodatkowe_wyposazenie
ADD CONSTRAINT faktura_detale_fk FOREIGN KEY ( kod_samochodu )
REFERENCES faktura_detale ( kod_samochodu )
ON DELETE CASCADE
NOT DEFERRABLE;

ALTER TABLE faktura_detale
ADD CONSTRAINT faktura_naglowek_fk FOREIGN KEY ( nr_faktury )
REFERENCES faktura_naglowek ( nr_faktury )
ON DELETE CASCADE
NOT DEFERRABLE;

ALTER TABLE faktura_naglowek
ADD CONSTRAINT klient_fk FOREIGN KEY ( id_klienta )
REFERENCES klient ( id_klienta )
NOT DEFERRABLE;

ALTER TABLE faktura_detale
ADD CONSTRAINT marka_fk FOREIGN KEY ( id_marki )
REFERENCES marka ( id_marki )
NOT DEFERRABLE;

ALTER TABLE model
ADD CONSTRAINT marka_fkv2 FOREIGN KEY ( id_marki )
REFERENCES marka ( id_marki )
ON DELETE CASCADE
NOT DEFERRABLE;

ALTER TABLE wybrane_dodatkowe_wyposazenie
ADD CONSTRAINT model_detale_fk FOREIGN KEY ( id_modelu,
                                                                id_wyposazenia )
REFERENCES model_detale ( id_modelu,
                                                                id_wyposazenia )
NOT DEFERRABLE;

ALTER TABLE faktura_detale
ADD CONSTRAINT model_fk FOREIGN KEY ( id_modelu )
REFERENCES model ( id_modelu )
NOT DEFERRABLE;

ALTER TABLE model_detale
ADD CONSTRAINT model_fkv2 FOREIGN KEY ( id_modelu )
REFERENCES model ( id_modelu )
ON DELETE CASCADE
NOT DEFERRABLE;

ALTER TABLE faktura_naglowek
ADD CONSTRAINT pracownik_fk FOREIGN KEY ( id_pracownika )
REFERENCES pracownik ( id_pracownika )
NOT DEFERRABLE;

ALTER TABLE model_detale
ADD CONSTRAINT wyposazenie_dodatkowe_fk FOREIGN KEY ( id_wyposazenia )
REFERENCES wyposazenie_dodatkowe ( id_wyposazenia )
ON DELETE CASCADE
NOT DEFERRABLE;

```

```

--tworzenie wyzwalaczy
create or replace TRIGGER tr_ins_faktura_detale
BEFORE INSERT ON faktura_detale
FOR EACH ROW
declare
v_cena_zakupu faktura_detale.cena_zakupu%type;

begin
select cena_sprzedazy into v_cena_zakupu from model where id_modelu = :new.id_modelu;
:new.kod_samochodu := seq_faktura_detale.nextval;
:new.cena_zakupu :=v_cena_zakupu;--dodanie do wartosci poczatkowej szczegolnego miejsca na fakturze wartosci modelu ktory zostal wybrany

update faktura_naglowek
set wartosc_faktury = wartosc_faktury+ v_cena_zakupu
where nr_faktury = :new.nr_faktury;--aktualizacja calej fatury

end;
/

-----
create or replace TRIGGER TR_INS_FAKTURA_NAGLOWEK_BEFORE BEFORE INSERT ON Faktura_naglowek
FOR EACH ROW
begin
:new.nr_faktury := seq_faktura_naglowek.nextval;
:new.wartosc_faktury :=0;--ustawiana jest poczatkowa wartosc faktury na 0 oraz przypisywany jest jej id za pomoca sekwencji
end;
/

-----
create or replace trigger tr_wybrane_dodatkowe_wyposazenie
after insert on wybrane_dodatkowe_wyposazenie
for each row
begin

update faktura_detale
set cena_zakupu=cena_zakupu + (select cena from model_detale where id_modelu=:new.id_modelu and id_wyposazenia=:new.id_wyposazenia)
where kod_samochodu=:new.kod_samochodu;--do wartosci pola w fakturze dodawana jest wartosc dodatkowego wyposazenia auta ktore zostalo wybrane

update faktura_naglowek
set wartosc_faktury = wartosc_faktury + (select cena from model_detale where id_modelu=:new.id_modelu and id_wyposazenia=:new.id_wyposazenia)
where nr_faktury = (select nr_faktury from faktura_detale where kod_samochodu=:new.kod_samochodu);--aktualizacja wartosci calej faktury

end;

-----
/

```

^Podczas dodawania nowej faktury do bazy danych jej początkowa wartość ustawiana jest na zero(**tr\_ins\_faktura\_detale**) oraz zwiększana wraz z każdym razem kiedy do faktury zostaje dodane nowe auto(**tr\_ins\_faktura\_detale**), oraz jego wyposażenie dodatkowe(**tr\_wybrane\_dodatkowe\_wyposazenie**) .

**Generator danych:**

```

create or replace PROCEDURE P_FAKTURA_DETAL AS

v_indeks numeric(1);
v_faktura_naglowek_id faktura_naglowek.nr_faktury%type;
v_id_marka marka.id_marka%type;
v_id_modelu model.id_modelu%type;
v_x numeric(1);

cursor cur_faktura_naglowek is
select nr_faktury from faktura_naglowek;

BEGIN

open cur_faktura_naglowek;

loop

fetch cur_faktura_naglowek into v_faktura_naglowek_id;--dla kazdej faktury losowane sa losowe pola z tabeli faktura_detale
exit when cur_faktura_naglowek%notfound;

-----
v_indeks:=0;
select dbms_random.value(2,6) into v_x from dual;--kazdej fakturze przypisywane jest od 2 do 6 detali
while v_indeks < v_x loop
select id_marka into v_id_marka from (select id_marka from marka order by dbms_random.value)
where rownum =1;

select id_modelu into v_id_modelu from(select id_modelu from model where id_marka=v_id_marka order by dbms_random.value)
where rownum=1;

insert into faktura_detale(nr_faktury,id_marka,id_modelu)--jako detale faktury wprowadzane sa wartosci wylosowane powyzej
values(v_faktura_naglowek_id,v_id_marka,v_id_modelu);--

v_indeks:=v_indeks+1;
end loop;
-----
end loop;
close cur_faktura_naglowek;

END P_FAKTURA_DETAL;

```

P\_faktura\_detale odpowiada za generowanie detali(samochodów) które zakupił klient-

Dla każdego nagłówka faktury umieszczonego w tabeli faktura\_detale generowane jest od 2 do 6 detali.

```

create or replace PROCEDURE P_FAKTURA_NAGLOWEK AS

v_id_klienta klient.id_klienta%type;
v_id_pracownika pracownik.id_pracownika%type;
v_data date;
v_indeks numeric(2) :=0;

BEGIN

loop
exit when v_indeks=20;--tworzone jest 20 rekordow w tabeli faktura_naglowek

select id_klienta into v_id_klienta from (select id_klienta from klient order by dbms_random.value)
where rownum=1;
select id_pracownika into v_id_pracownika from (select id_pracownika from pracownik order by dbms_random.value)
where rownum=1;

select (sysdate-(select dbms_random.value(0,600) from dual)) into v_data from dual;

insert INTO faktura_naglowek(id_pracownika,id_klienta,data_faktury)--do tabeli faktura_naglowek wprowadzane zostaja dane wylosowane powyzej
values(v_id_pracownika,v_id_klienta,v_data);
v_indeks := v_indeks+1;

end loop;

END P_FAKTURA_NAGLOWEK;

```

^ p\_faktura\_naglowek generuje 20 nagłówków faktur na podstawie początkowych danych umieszczonych w tabelach .

```

create or replace procedure p_model_detale
as
v_id_marka marka.id_marka%type;--konkretne dodatkowe wyposazenia sa przewidziane dla konkretnych modeli(nie kazdy model musi miec te same dodatkowe wyposazenia)
cursor cur_marka is
select id_marka from marka;

begin
    open cur_marka;

    loop
        fetch cur_marka into v_id_marka;
        exit when cur_marka%notfound;
        p_model_detale_2(v_id_marka);--tutaj wywolwana jest czesc losujaca wyposazenie dla modelu
    end loop;

    close cur_marka;

end;

/

```

P\_model\_detale przekazuje kolejno każdą markę samochodu z tabeli MARKA do procedury

P\_model\_detale\_2 aby w niej dokonać przypisu dostępnych dodatkowych wyposażań dla modeli danej marki.

```

create or replace PROCEDURE p_model_detale_2
(v_id_marka marka.id_marka%type)

AS
v_i numeric(1) :=1;
v_id_modelu model.id_modelu%type;
v_id_wyposazenia wyposazenie_dodatkowe.id_wyposazenia%type;
v_cena_wyposazenia wyposazenie_dodatkowe.koszt_producenta%type;
v_$ wyposazenie_dodatkowe.koszt_producenta%type;

cursor cur_model is
select id_modelu from model
where id_marka = v_id_marka;

cursor cur_wyp_dodat is select id_wyposazenia, koszt_producenta
from (select id_wyposazenia, koszt_producenta from wyposazenie_dodatkowe
order by dbms_random.value) where rownum<=5;

BEGIN

open cur_model ;

loop--PETLA LOSUJACA model SAMOCHODu

fetch cur_model into v_id_modelu;
exit when cur_model%notfound;

-----
open cur_wyp_dodat;

loop--dla kazdego modelu losowane jest 5 roznych opcji wyposazenia dodatkowego

select round(dbms_random.value(3000,5000),0) into v_$ from dual;-- generowany jest tutaj narzut na wyposazenie dodatkowe

fetch cur_wyp_dodat into v_id_wyposazenia,v_cena_wyposazenia;
exit when cur_wyp_dodat%notfound;
v_cena_wyposazenia := v_cena_wyposazenia+v_$;

insert into model_detale
values(v_id_modelu,v_id_wyposazenia,v_cena_wyposazenia);--wprowadzanie wylosowanych powyzej wartosci

end loop;

close cur_wyp_dodat;
-----

end loop;

close cur_model;

END p_model_detale_2;

```

Powyzsza procedura przyporządkowuje każdemu modelowi każdej marki 5 możliwych dodatkowych wyposażań jakie jest może nabyć klient podczas zakupu auta.

```

create or replace PROCEDURE P_WYBRANE_DODATKOWE_WYPOSAZENIE AS --procedura ta odpowiada za wylosowanie wybranych wyposazen dodatkowych dla klientow

v_x numeric(1):=0;
v_ilosc numeric(1);

v_id_modelu model.id_modelu%type;
v_kod_samochodu faktura_detale.kod_samochodu%type;
v_id_wyposazenia wyposazenie_dodatkowe.id_wyposazenia%type;

cursor cur_uniwersalna is
select kod_samochodu, id_modelu
from faktura_detale;

-----
cursor cur_id_wyposazenia is
select id_wyposazenia from model_detale
where id_modelu=v_id_modelu
order by dbms_random.value;

BEGIN

    open cur_uniwersalna;
loop
    fetch cur_uniwersalna into v_kod_samochodu,v_id_modelu;
    exit when cur_uniwersalna%notfound;

    -----
    v_x:=0;
    select dbms_random.value(0,5) into v_ilosc from dual;--tutaj losowana jest ilosc wyposazen wybranych przez klienta
    open cur_id_wyposazenia;

    while v_x < v_ilosc loop-- w tej petli nastepuje losowanie dodatkowych wyposazen w ilosci wylosowanej wzczesniej
    fetch cur_id_wyposazenia into v_id_wyposazenia;

    insert into wybrane_dodatkowe_wyposazenie--wprowadzanie
    values(v_id_modelu,v_id_wyposazenia,v_kod_samochodu);

    v_x:=v_x+1;
    end loop;
    close cur_id_wyposazenia;

    -----

    end loop;
    close cur_uniwersalna;

END P_WYBRANE_DODATKOWE_WYPOSAZENIE;

```

ad Querv Language file

length : 20 292

Powyższa procedura generuje przypadkowe wybory klientów umieszczonych w bazie.

Każdy klient może wybrać wszystkie dodatkowe wyposażenia lub nie wybrać ich wcale.



```
--wprowadzanie danych staych na których bazowane jest losowanie
```

```
insert into marka  
values(1,'BMW');
```

```
insert into marka  
values(2,'Porsche');
```

```
insert into marka  
values(3,'Audi');
```

```
--Uzupełnianie modeli BMW-----
```

```
insert into model  
values(seq_model.nextval,1,'seria 1',200000,250000);
```

```
insert into model  
values(seq_model.nextval,1,'seria 2',250000,300000);
```

```
insert into model  
values(seq_model.nextval,1,'seria 3',300000,350000);
```

```
insert into model  
values(seq_model.nextval,1,'seria 4',350000,400000);
```

```
insert into model  
values(seq_model.nextval,1,'seria 5',400000,450000);
```

```
--uzupenianie porsche-----
```

```
insert into model  
values(seq_model.nextval,2,'Cayman',200000,250000);
```

```
insert into model  
values(seq_model.nextval,2,'Boxster',250000,300000);
```

```
insert into model  
values(seq_model.nextval,2,'Macan',300000,350000);
```

```
insert into model  
values(seq_model.nextval,2,'Panamera',350000,400000);
```

```
insert into model  
values(seq_model.nextval,2,'911',400000,450000);
```

```

}--uzupełnianie modeli audi-----
insert into model
values(seq_model.nextval,3,'A3',200000,250000);

insert into model
values(seq_model.nextval,3,'A4',250000,300000);

insert into model
values(seq_model.nextval,3,'A5',300000,350000);

insert into model
values(seq_model.nextval,3,'A6',350000,400000);

insert into model
values(seq_model.nextval,3,'A7',400000,450000);

--uzupełnianie klientow -----

insert into klient
values(seq_klient.nextval,'Karolina','Cybulska');

insert into klient
values(seq_klient.nextval,'Wiktor','Urbański');

insert into klient
values(seq_klient.nextval,'Adrian','Łukasik');

insert into klient
values(seq_klient.nextval,'Paweł','Pawowski');

insert into klient
values(seq_klient.nextval,'Kacper','Górski');

insert into klient
values(seq_klient.nextval,'Zuzanna','Kowalska');

insert into klient
values(seq_klient.nextval,'Julia','Nowicka');

insert into klient
values(seq_klient.nextval,'Katarzyna','Kowalska');

insert into klient
values(seq_klient.nextval,'Karina','Nowakowska');

--uzupełnianie pracowników-----

insert into pracownik
values(seq_pracownik.nextval,'Filip','Kwiatkowski');

insert into pracownik
values(seq_pracownik.nextval,'Julia','Żak');

insert into pracownik
values(seq_pracownik.nextval,'Laura','Dębrowska');

insert into pracownik
values(seq_pracownik.nextval,'Olaf','Kopeć');

```

```

--uzupełnianie dodatkowego wyposażenia-----

insert into wyposażenie_dodatkowe
values (SEQ_WYP_DODAT.nextval, 'ABS', SEQ_WYP_DODAT_KOSZT.nextval);

insert into wyposażenie_dodatkowe
values (SEQ_WYP_DODAT.nextval, 'Centralny zamek', SEQ_WYP_DODAT_KOSZT.nextval);

insert into wyposażenie_dodatkowe
values (SEQ_WYP_DODAT.nextval, 'Elektryczne szyby przednie', SEQ_WYP_DODAT_KOSZT.nextval);

insert into wyposażenie_dodatkowe
values (SEQ_WYP_DODAT.nextval, 'Immobilizer', SEQ_WYP_DODAT_KOSZT.nextval);

insert into wyposażenie_dodatkowe
values (SEQ_WYP_DODAT.nextval, 'ASR', SEQ_WYP_DODAT_KOSZT.nextval);

insert into wyposażenie_dodatkowe
values (SEQ_WYP_DODAT.nextval, 'Asystent parkowania', SEQ_WYP_DODAT_KOSZT.nextval);

insert into wyposażenie_dodatkowe
values (SEQ_WYP_DODAT.nextval, 'Asystent pasa ruchu', SEQ_WYP_DODAT_KOSZT.nextval);

insert into wyposażenie_dodatkowe
values (SEQ_WYP_DODAT.nextval, 'Bluetooth', SEQ_WYP_DODAT_KOSZT.nextval);

insert into wyposażenie_dodatkowe
values (SEQ_WYP_DODAT.nextval, 'Czujnik deszczu', SEQ_WYP_DODAT_KOSZT.nextval);

insert into wyposażenie_dodatkowe
values (SEQ_WYP_DODAT.nextval, 'Klimatyzacja dwustrefowa', SEQ_WYP_DODAT_KOSZT.nextval);

/

BEGIN
P_MODEL_DETALE;
END;
/

BEGIN
P_FAKTURA_NAGLOWEK;
END;
/

BEGIN
P_FAKTURA_DETALE;
END;
/

BEGIN
P_WYBRANE_DODATKOWE_WYPOSAZENIE;
END;
/

```

## Tworzenie perspektyw:

```
CREATE OR REPLACE VIEW FAKTURA AS
select fn.nr_faktury,data_faktury,wartosc_faktury,fd.kod_samochodu,cena_zakupu,nazwa_marki,nazwa_modelu,cena_sprzedazy "cena bazowa",nazwa_wyposazenia,cena
from faktura_naglowek fn
join faktura_detale fd on fd.nr_faktury=fn.nr_faktury
join marka m on m.id_marki=fd.id_marki
join model mod on mod.id_modelu=fd.id_modelu
left outer join wybrane_dodatkowe_wyposazenie wdw on wdw.kod_samochodu=fd.kod_samochodu
left outer join model_detale md on md.id_modelu=wdw.id_modelu and md.id_wyposazenia=wdw.id_wyposazenia
left outer join wyposazenie_dodatkowe wd on wd.id_wyposazenia=md.id_wyposazenia
order by kod_samochodu;
```

```
CREATE OR REPLACE VIEW MIESIECZNE_ZESTAWIENIE_SPRZEDAZY_MODELI AS
select EXTRACT(YEAR FROM data_faktury)"rok",EXTRACT(month FROM data_faktury)"miesiac",nazwa_marki,nazwa_modelu,count(kod_samochodu)"ilosc sprzedanych" from faktura_detale fd
join marka m on m.id_marki=fd.id_marki
join model mod on mod.id_modelu=fd.id_modelu
join faktura_naglowek fn on fn.nr_faktury=fd.nr_faktury
group by nazwa_marki,nazwa_modelu,EXTRACT(YEAR FROM data_faktury),EXTRACT(month FROM data_faktury)
order by "rok","miesiac","ilosc sprzedanych";
```

```
CREATE OR REPLACE VIEW ROCZNE_ZESTAWIENIE_SPRZEDAZY_MODELI AS
select EXTRACT(YEAR FROM data_faktury)"rok",nazwa_marki,nazwa_modelu,count(kod_samochodu)"ilosc sprzedanych" from faktura_detale fd
join marka m on m.id_marki=fd.id_marki
join model mod on mod.id_modelu=fd.id_modelu
join faktura_naglowek fn on fn.nr_faktury=fd.nr_faktury
group by nazwa_marki,nazwa_modelu,EXTRACT(YEAR FROM data_faktury)
order by "rok" asc,"ilosc sprzedanych" desc;
```

```
CREATE OR REPLACE VIEW ZESTAWIENIE_KOSZTOW_ZAMOWIENIA AS
select data_faktury,fd.kod_samochodu,cena_produkcji "koszty prod",nvl(sum(wd.koszt_producenta),0)"koszty wyp",cena_zakupu,cena_produkcji+nvl(sum(wd.koszt_producenta),0) as "koszty" from faktura_detale fd
join model mod on fd.id_modelu=mod.id_modelu
left outer join wybrane_dodatkowe_wyposazenie wdw on wdw.kod_samochodu=fd.kod_samochodu
left outer join model_detale md on md.id_modelu=wdw.id_modelu and wdw.id_wyposazenia=md.id_wyposazenia
left outer join wyposazenie_dodatkowe wd on wd.id_wyposazenia=md.id_wyposazenia
left outer join faktura_naglowek fn on fn.nr_faktury=fd.nr_faktury
group by fd.kod_samochodu,cena_zakupu,cena_produkcji,data_faktury
order by fd.kod_samochodu;
```

```
CREATE OR REPLACE VIEW ZYSKI_MIESIECZNE AS
select extract(year from data_faktury)"rok",extract(month from data_faktury)"miesiac",sum(cena_zakupu-"koszty")"zyski" from zestawienie_kosztow_zamowienia
group by extract(year from data_faktury),extract(month from data_faktury)
order by "rok","miesiac","zyski";
```

```
CREATE OR REPLACE VIEW ZYSKI_ROCZNE AS
select extract(year from data_faktury)"rok",sum(cena_zakupu-"koszty")"zyski" from zestawienie_kosztow_zamowienia
group by extract(year from data_faktury)
order by "rok","zyski";
```

Opis efektów działania perspektyw zostanie tu pominięty ze względu na prostotę sprawdzenia tego.

Skrypt Usuający bazę:

```
DROP TABLE faktura_detale CASCADE CONSTRAINTS;
DROP TABLE faktura_naglowek CASCADE CONSTRAINTS;
DROP TABLE klient CASCADE CONSTRAINTS;
DROP TABLE marka CASCADE CONSTRAINTS;
DROP TABLE model CASCADE CONSTRAINTS;
DROP TABLE model_detale CASCADE CONSTRAINTS;
DROP TABLE pracownik CASCADE CONSTRAINTS;
DROP TABLE wybrane_dodatkowe_wyposazenie CASCADE CONSTRAINTS;
DROP TABLE wyposazenie_dodatkowe CASCADE CONSTRAINTS;

DROP SEQUENCE seq_faktura_detale;
DROP SEQUENCE seq_faktura_naglowek;
DROP SEQUENCE SEQ_KLIENT;
DROP SEQUENCE SEQ_MODEL;
DROP SEQUENCE SEQ_PRACOWNIK;
DROP SEQUENCE SEQ_WYP_DODAT;
DROP SEQUENCE SEQ_WYP_DODAT_KOSZT;

drop view MIESIECZNE_ZESTAWIENIE_SPRZEDAZY_MODELI;
drop view FAKTURA;
drop view LACZNIE_KLIENCI;
drop view NAJLEPSZY_PRACOWNIK;
drop view ROCZNE_ZESTAWIENIE_SPRZEDAZY_MODELI;
drop view ZESTAWIENIE_KOSZTOW_ZAMOWIENIA;
drop view ZYSKI_MIESIECZNE;
drop view ZYSKI_ROCZNE;

drop trigger tr_ins_faktura_detale;
drop trigger TR_INS_FAKTURA_NAGLOWEK_BEFORE;
drop trigger tr_wybrane_dodatkowe_wyposazenie;
drop procedure P_FAKTURA_DETALE;
drop procedure P_FAKTURA_NAGLOWEK;
drop procedure p_model_detale;
drop procedure p_model_detale_2;
drop procedure P_WYBRANE_DODATKOWE_WYPOSAZENIE;
```

Instalacja:

W celu instalacji bazy danych wystarczy zaimportować a następnie uruchomić skrypt Projekt.ddl

UWAGA:

Skrypt służy do jednorazowej instalacji bazy danych salonu samochodowego. **NIE** wskazane jest używanie ponowne zaimplementowanych w nim procedur odpowiedzialnych za generowanie danych. O ile ponowne używanie procedur generujących dane losowe dla tabel faktura\_nagłówki oraz faktura\_detale nie będą miały znaczącego wpływu na poprawność danych w tabeli o tyle ponowne użycie procedury generującej dane dla tabeli wybrane\_dodatkowe\_wyposażenie prowadzić będzie do zdublowania niektórych wartości dla danej pozycji (to tak jakbyśmy zamówili sobie 2xabs dla jednego auta) przez co wartość faktury automatycznie wzrośnie. Procedura ta bazuje na autach znajdujących się już w bazie.

## Zestawienia wygenerowane w programie Jasper Reports



23/03/2019

### Zestawienie zysków z miesięcy

Rok	Miesiac	Zyski
2017	9	321102
2017	10	935455
2017	12	465480
2018	2	502432
2018	3	995601
2018	4	130039
2018	5	156771
2018	6	226813
2018	8	526776
2018	9	186035
2019	2	322871

Ilość rekordów 11



## ZESTAWIENIE MIESIĘCZNYCH SPRZEDAŻY

Rok	Miesiąc	Marka	Model	Ilość
2017	9	BMW	seria 1	1
2017	9	Porsche	911	1
2017	9	Audi	A4	1
2017	9	BMW	seria 2	1
2017	9	Porsche	Panamera	1
2017	10	Porsche	911	1
2017	10	Audi	A3	1
2017	10	BMW	seria 4	1
2017	10	Audi	A5	1
2017	10	Porsche	Macan	2
2017	10	Audi	A7	2
2017	10	BMW	seria 1	2
2017	10	Porsche	Boxster	3
2017	10	BMW	seria 5	3
2017	12	BMW	seria 4	1
2017	12	Audi	A3	1