BAZY DANYCH

PROJEKT

Sklep z elektroniką

Wykonanie: Beata Dziewulska

Grupa: WCY19IY5S1

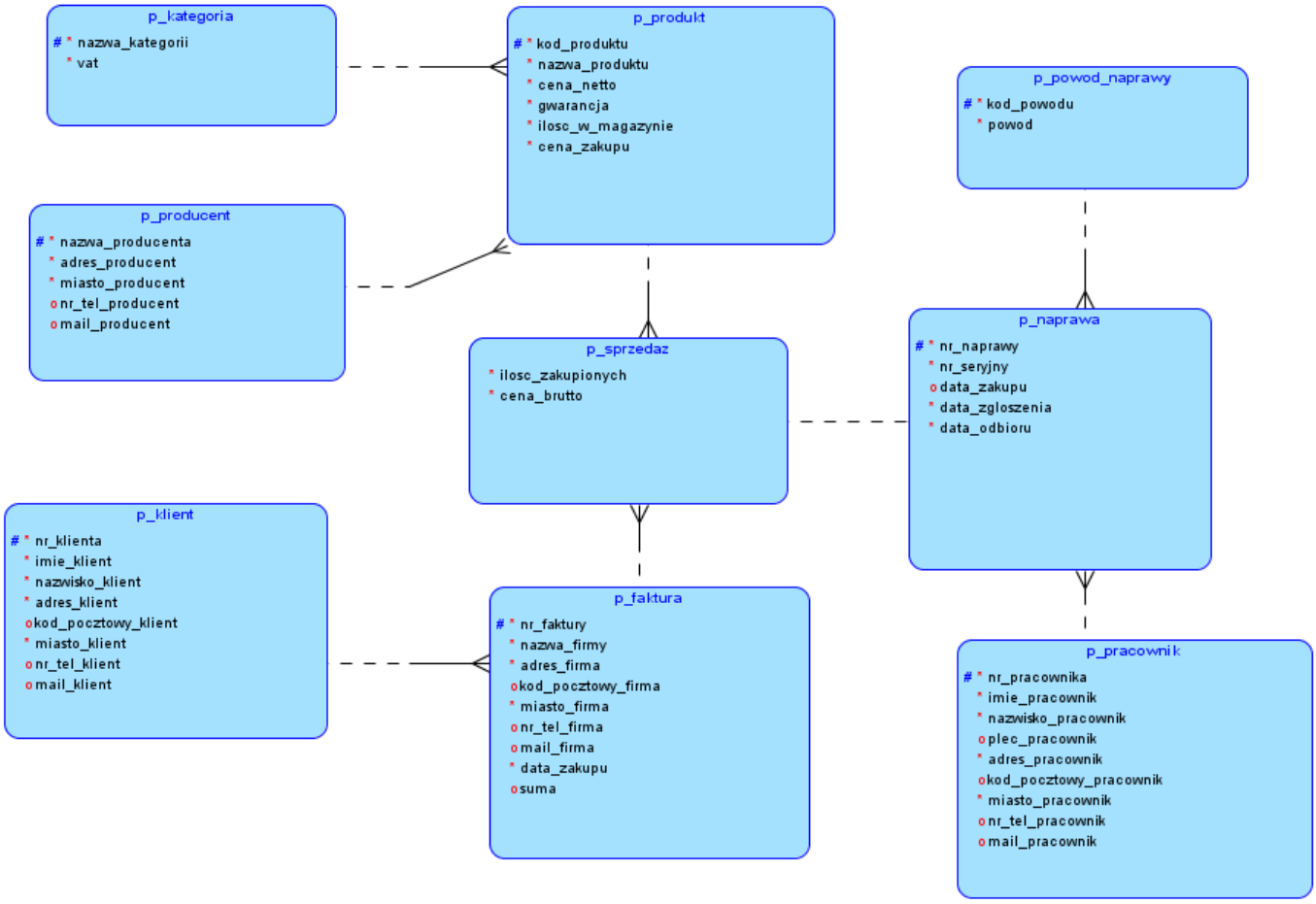
Data wykonania: 31.01.2021

Spis treści:

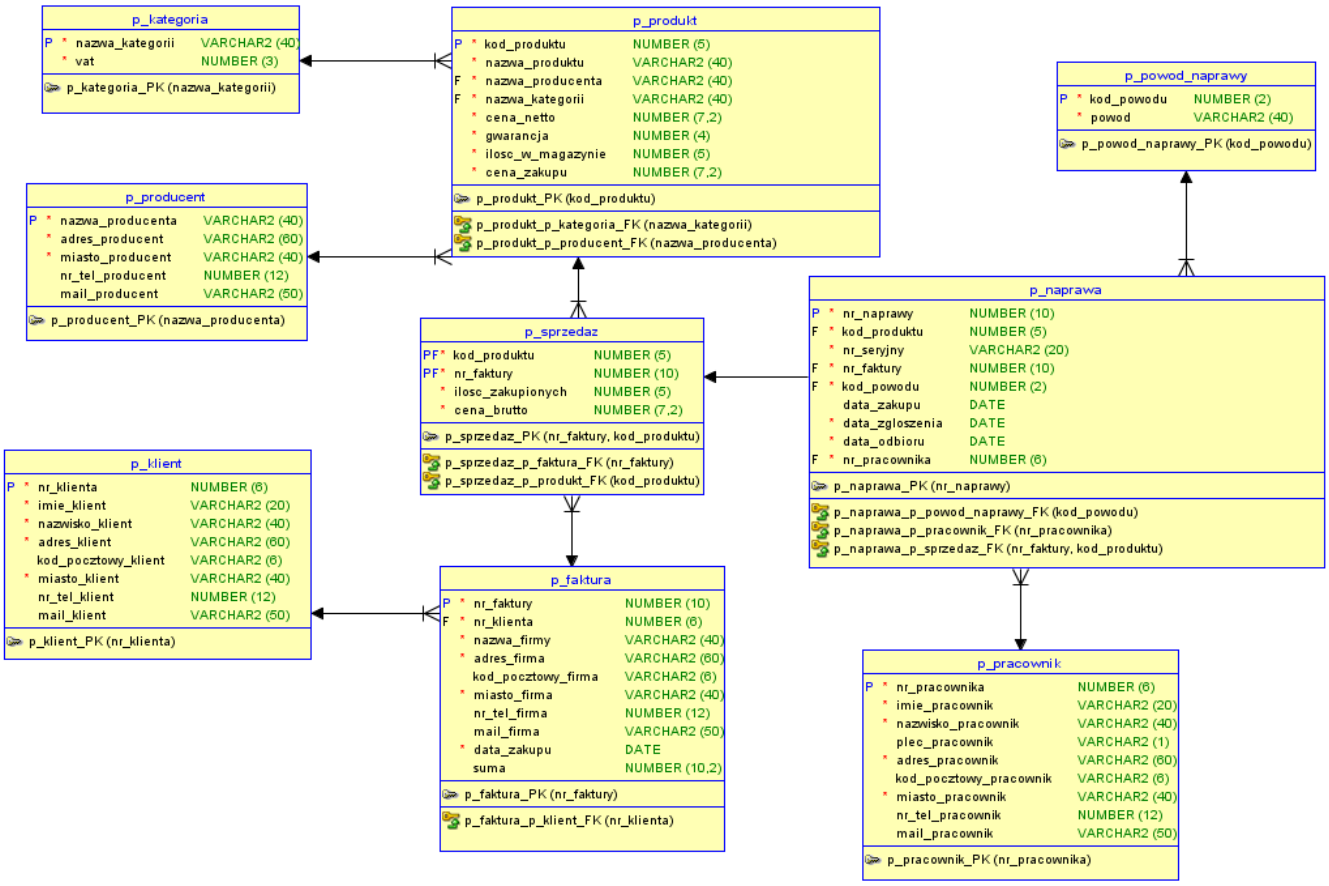
1. Analiza biznesowa

Rodzinna firma „revolution” otworzyła swój pierwszy sklep z elektroniką w Warszawie oraz sklep internetowy. Aby móc się rozwijać, firma potrzebuje bazy danych, która umożliwi przechowywanie danych klientów, zrealizowanych transakcji oraz informacji o produktach znajdujących się w magazynie. Baza danych powinna automatycznie aktualizować stan produktów, dokonywać potrzebnych obliczeń, które znacznie przyspieszą pracę oraz tworzyć zestawienia danych, które przydadzą się do m. in. obliczania miesięcznego i rocznego przychodu, planowania dostaw produktów, wystawiana faktur. W sklepie stacjonarnym znajduje się też serwis. Pracownicy sklepu muszą więc mieć dostęp do wystawionych faktur, aby móc potwierdzić, że dany produkt pochodzi z ich sklepu oraz sprawdzić czy produkt jest objęty gwarancją. Klient powinien wiedzieć kiedy jego naprawiony produkt będzie gotowy do obioru. Każdy produkt powinien mieć kod, nazwę produktu, nazwę producenta, kategorię, cenę netto, gwarancję, informację o ilości sztuk jaka została w magazynie oraz cenie za jaką sklep kupił dany produkt (aby móc obliczyć zysk). Faktura powinna zawierać numer faktury, dane klienta, dane firmy, datę sprzedaży oraz wartość faktury. W bazie powinny znajdować się dane klientów oraz pracowników takie jak godność, adres, numer telefonu, adres mailowy. Należy też przechowywać informacje o naprawie tj. numer naprawy, informacje o produkcie, dane z faktury zakupu, datę zgłoszenia do naprawy oraz datę odbioru. Naprawa jest możliwa tylko jeśli produkt jest objęty gwarancją. Baza ma umożliwić sprawdzenie m. in. który produkt najlepiej i najgorzej się sprzedaje, jaki jest najbardziej wadliwy produkt, który klient zakupił najwięcej produktów.

2. Model logiczny



3. Model relacyjny



4. Tworzenie

W pliku o nazwie projekt_create.ddl znajdują się zdania umożliwiające stworzenie całej struktury wraz z niezbędnymi modyfikacjami:

```
CREATE TABLE p_faktura (
   nr_faktury         NUMBER(10) NOT NULL,
   nr_klienta         NUMBER(6) NOT NULL,
   nazwa_firmy        VARCHAR2(40) NOT NULL,
   adres_firma        VARCHAR2(60) NOT NULL,
   kod_pocztowy_firma  VARCHAR2(6),
   miasto_firma       VARCHAR2(40) NOT NULL,
   nr_tel_firma       NUMBER(12),
   mail_firma         VARCHAR2(50),
   data_zakupu        DATE NOT NULL,
   suma               NUMBER(10, 2)
);

ALTER TABLE p_faktura ADD CONSTRAINT p_faktura_pk PRIMARY KEY ( nr_faktury );

CREATE TABLE p_kategoria (
   nazwa_kategorii  VARCHAR2(40) NOT NULL,
   vat              NUMBER(3) NOT NULL
);

ALTER TABLE p_kategoria ADD CONSTRAINT p_kategoria_pk PRIMARY KEY ( nazwa_kategorii
);

CREATE TABLE p_klient (
   nr_klienta         NUMBER(6) NOT NULL,
   imie_klient        VARCHAR2(20) NOT NULL,
   nazwisko_klient    VARCHAR2(40) NOT NULL,
```

```
   adres_klient        VARCHAR2(60) NOT NULL,
   kod_pocztowy_klient  VARCHAR2(6),
   miasto_klient       VARCHAR2(40) NOT NULL,
   nr_tel_klient       NUMBER(12),
   mail_klient         VARCHAR2(50)
);

ALTER TABLE p_klient ADD CONSTRAINT p_klient_pk PRIMARY KEY ( nr_klienta );

CREATE TABLE p_naprawa (
   nr_naprawy      NUMBER(10) NOT NULL,
   kod_produktu    NUMBER(5) NOT NULL,
   nr_seryjny      VARCHAR2(20) NOT NULL,
   nr_faktury      NUMBER(10) NOT NULL,
   kod_powodu      NUMBER(2) NOT NULL,
   data_zakupu     DATE,
   data_zgloszenia  DATE NOT NULL,
   data_odbioru    DATE NOT NULL,
   nr_pracownika   NUMBER(6) NOT NULL
);

ALTER TABLE p_naprawa ADD CONSTRAINT p_naprawa_pk PRIMARY KEY ( nr_naprawy );

CREATE TABLE p_powod_naprawy (
   kod_powodu  NUMBER(2) NOT NULL,
   powod       VARCHAR2(40) NOT NULL
);

ALTER TABLE p_powod_naprawy ADD CONSTRAINT p_powod_naprawy_pk PRIMARY KEY (
kod_powodu );
```

```
CREATE TABLE p_pracownik (
   nr_pracownika          NUMBER(6) NOT NULL,
   imie_pracownik         VARCHAR2(20) NOT NULL,
   nazwisko_pracownik     VARCHAR2(40) NOT NULL,
   plec_pracownik         VARCHAR2(1),
   adres_pracownik        VARCHAR2(60) NOT NULL,
   kod_pocztowy_pracownik  VARCHAR2(6),
   miasto_pracownik       VARCHAR2(40) NOT NULL,
   nr_tel_pracownik       NUMBER(12),
   mail_pracownik         VARCHAR2(50)
);

ALTER TABLE p_pracownik ADD CONSTRAINT p_pracownik_pk PRIMARY KEY (
nr_pracownika );

CREATE TABLE p_producent (
   nazwa_producenta  VARCHAR2(40) NOT NULL,
   adres_producent   VARCHAR2(60) NOT NULL,
   miasto_producent  VARCHAR2(40) NOT NULL,
   nr_tel_producent  NUMBER(12),
   mail_producent    VARCHAR2(50)
);

ALTER TABLE p_producent ADD CONSTRAINT p_producent_pk PRIMARY KEY (
nazwa_producenta );

CREATE TABLE p_produkt (
   kod_produktu      NUMBER(5) NOT NULL,
   nazwa_produktu    VARCHAR2(40) NOT NULL,
```

```
   nazwa_producenta   VARCHAR2(40) NOT NULL,
   nazwa_kategorii    VARCHAR2(40) NOT NULL,
   cena_netto         NUMBER(7, 2) NOT NULL,
   gwarancja          NUMBER(4) NOT NULL,
   ilosc_w_magazynie  NUMBER(5) NOT NULL,
         cena_zakupu      NUMBER(7,2) NOT NULL
);

ALTER TABLE p_produkt ADD CONSTRAINT p_produkt_pk PRIMARY KEY ( kod_produktu );

CREATE TABLE p_sprzedaz (
   kod_produktu       NUMBER(5) NOT NULL,
   nr_faktury         NUMBER(10) NOT NULL,
   ilosc_zakupionych  NUMBER(5) NOT NULL,
   cena_brutto        NUMBER(7, 2) NOT NULL
);

ALTER TABLE p_sprzedaz ADD CONSTRAINT p_sprzedaz_pk PRIMARY KEY ( nr_faktury,
                                                                  kod_produktu );

ALTER TABLE p_faktura
   ADD CONSTRAINT p_faktura_p_klient_fk FOREIGN KEY ( nr_klienta )
      REFERENCES p_klient ( nr_klienta );

ALTER TABLE p_naprawa
   ADD CONSTRAINT p_naprawa_p_powod_naprawy_fk FOREIGN KEY ( kod_powodu )
      REFERENCES p_powod_naprawy ( kod_powodu );

ALTER TABLE p_naprawa
```

```
   ADD CONSTRAINT p_naprawa_p_pracownik_fk FOREIGN KEY ( nr_pracownika )
      REFERENCES p_pracownik ( nr_pracownika );

ALTER TABLE p_naprawa
   ADD CONSTRAINT p_naprawa_p_sprzedaz_fk FOREIGN KEY ( nr_faktury,
                                        kod_produktu )
      REFERENCES p_sprzedaz ( nr_faktury,
                        kod_produktu );

ALTER TABLE p_produkt
   ADD CONSTRAINT p_produkt_p_kategoria_fk FOREIGN KEY ( nazwa_kategorii )
      REFERENCES p_kategoria ( nazwa_kategorii );

ALTER TABLE p_produkt
   ADD CONSTRAINT p_produkt_p_producent_fk FOREIGN KEY ( nazwa_producenta )
      REFERENCES p_producent ( nazwa_producenta );

ALTER TABLE p_sprzedaz
   ADD CONSTRAINT p_sprzedaz_p_faktura_fk FOREIGN KEY ( nr_faktury )
      REFERENCES p_faktura ( nr_faktury );

ALTER TABLE p_sprzedaz
   ADD CONSTRAINT p_sprzedaz_p_produkt_fk FOREIGN KEY ( kod_produktu )
      REFERENCES p_produkt ( kod_produktu );


alter table P_Faktura
modify nazwa_firmy default 'revolution';
--adres
```

```
alter table P_Faktura
modify adres_firma default 'Złota 18';
--kod pocztowy
alter table P_Faktura
modify kod_pocztowy_firma default '01-908';
--miasto
alter table P_Faktura
modify miasto_firma default 'Warszawa';
--nr tel
alter table P_Faktura
modify nr_tel_firma default '521361666';
--mail
alter table P_Faktura
modify mail_firma default 'revolution@gmail.com';
--suma
alter table p_faktura
modify suma default 0;

--DATA FAKTURY
alter table P_Faktura
modify data_zakupu default to_date(SYSDATE, 'DD.MM.YYYY');

--P_NAPRAWA:
--data zgloszenia
alter table P_Naprawa
modify data_zgloszenia default to_date(SYSDATE, 'DD.MM.YYYY');

alter table p_produkt add constraint ilosc_w_magazynie_check check (ilosc_w_magazynie
>= 0);
```

```
--SEKWENCJE:
--KOD PRODUKTU
CREATE SEQUENCE P_SEQ_KOD_PRODUKTU
minvalue 1
maxvalue 99999999999999999999
start with 1
increment by 1
cache 20;
--NR FAKTURY
CREATE SEQUENCE P_SEQ_NR_FAKTURY
minvalue 1
maxvalue 99999999999999999999
start with 1
increment by 1
cache 20;
--NR KLIENTA
CREATE SEQUENCE P_SEQ_NR_KLIENTA
minvalue 1
maxvalue 99999999999999999999
start with 1
increment by 1
cache 20;
--NR NAPRAWY
CREATE SEQUENCE P_SEQ_NR_NAPRAWY
minvalue 1
maxvalue 99999999999999999999
start with 1
increment by 1
cache 20;
```

```
--FUNKCJE

--Do kiedy wazna jest gwarancja

create or replace FUNCTION DO_KIEDY_GWARANCJA (kod p_produkt.kod_produktu%type,
data_z p_faktura.data_zakupu%type, faktura p_faktura.nr_faktury%type)

RETURN DATE AS

do_kiedy date;

BEGIN

   select Add_months(data_z,gwarancja) into do_kiedy

   from p_faktura f

   join p_sprzedaz s on s.nr_faktury = f.nr_faktury

   join p_produkt p on p.kod_produktu = s.kod_produktu

   where p.kod_produktu = kod and f.nr_faktury = faktura;

return do_kiedy;

END DO_KIEDY_GWARANCJA;

/

--Czy gwarancja jest jeszcze wazna

create or replace FUNCTION CZY_GWARANCJA_WAZNA (kod
p_produkt.kod_produktu%type, faktura p_faktura.nr_faktury%type, data_zgl
p_naprawa.data_zgloszenia%type)

RETURN BOOLEAN AS

data_wygasniecia date;

data_zak date;

BEGIN


   select data_zakupu into data_zak

   from p_faktura f

   join p_sprzedaz s on f.nr_faktury = s.nr_faktury

   join p_produkt p on p.kod_produktu = s.kod_produktu

   where s.kod_produktu = kod and s.nr_faktury = faktura;
```

```
   data_wygasniecia := do_kiedy_gwarancja(kod, data_zak, faktura);

   if data_wygasniecia >= data_zgl then
   return true;
   else
   return false;
   end if;

END CZY_GWARANCJA_WAZNA;
/
--Liczenie ceny brutto
create or replace FUNCTION LICZ_VAT (kod p_sprzedaz.kod_produktu%type)
RETURN NUMBER AS
policzone number(7,2);
BEGIN
   select cena_netto+(cena_netto*vat/100) into policzone
   from p_produkt p
   join p_kategoria k on k.nazwa_kategorii = p.nazwa_kategorii
   where kod_produktu=kod;
   return policzone;
END LICZ_VAT;
/
--Liczenie ceny netto
create or replace FUNCTION LICZ_NETTO(cena p_produkt.cena_zakupu%type)
RETURN NUMBER AS
wynik number(7,2);
BEGIN
   wynik := cena + cena*0.15;
```

```
   return wynik;
END LICZ_NETTO;
/
--Liczenie zysku
create or replace FUNCTION ZYSK (cena p_produkt.cena_netto%type, ilosc
p_sprzedaz.ilosc_zakupionych%type, cena_z p_produkt.cena_zakupu%type)
RETURN NUMBER AS
zysk_s number(7,2);
BEGIN
   zysk_s := (cena - cena_z)*ilosc;
   return zysk_s;
END ZYSK;
/
--Ustawianie daty faktury
create or replace FUNCTION USTAW_DATE(faktura p_faktura.nr_faktury%type)
RETURN DATE AS
data_nowa date;
BEGIN
 select data_zakupu into data_nowa from p_faktura
 where nr_faktury = faktura;
 return data_nowa;
END USTAW_DATE;
/
--Czy produkt jest w magazynie
CREATE OR REPLACE FUNCTION CZY_JEST_W_MAGAZYNIE (kod
p_produkt.kod_produktu%type, ile_stare number, ile_nowe number)
RETURN NUMBER AS
ilosc_przed number;
ilosc_po number;
BEGIN
```

```
   select ilosc_w_magazynie into ilosc_przed from p_produkt
   where kod_produktu = kod;

   ilosc_po := ilosc_przed + ile_stare - ile_nowe;

   return ilosc_po;
END CZY_JEST_W_MAGAZYNIE;
/

--PROCEDURY
--Dostawa produktow
create or replace PROCEDURE DOSTAWA(kod in p_produkt.kod_produktu%type, ilosc in
p_produkt.ilosc_w_magazynie%type)
AS
BEGIN
   update p_produkt
   set ilosc_w_magazynie = ilosc_w_magazynie + ilosc
   where kod_produktu = kod;
   dbms_output.put_line('Dostarczono '||ilosc||' sztuk produktu');
END DOSTAWA;
/
--Oddawanie produktu do naprawy
create or replace PROCEDURE ODDAJ_DO_NAPRAWY(kod in
p_naprawa.kod_produktu%type, nr_s in p_naprawa.nr_seryjny%type,
nr_f in p_naprawa.nr_faktury%type, kod_p in p_naprawa.kod_powodu%type, data_zgl in
p_naprawa.data_zgloszenia%type,
nr_p in p_naprawa.nr_pracownika%type)
AS
data_odb date;
BEGIN
```

```
   if czy_gwarancja_wazna(kod, nr_f, data_zgl) = TRUE then
   insert into p_naprawa(kod_produktu, nr_seryjny, nr_faktury, kod_powodu,
data_zgloszenia, nr_pracownika)
   values(kod, nr_s, nr_f, kod_p, data_zgl, nr_p);
   dbms_output.put_line('Produkt nr '||kod||' jest objety gwarancja');
   dbms_output.put_line('Oddano produkt o numerze seryjnym '||nr_s||' do naprawy.
Bedzie gotowy do odbioru '||do_kiedy_gwarancja(kod, data_zgl, nr_f)||'. ');
   elsif czy_gwarancja_wazna(kod, nr_f, data_zgl) = FALSE then
   dbms_output.put_line('Gwarancja na produkt nr '||kod||' wygasla');
   end if;
END ODDAJ_DO_NAPRAWY;
/
--Rabat
create or replace PROCEDURE RABAT(kat p_kategoria.nazwa_kategorii%type, rabat_p
number)
AS
cena_stara number(7,2);
cena_nowa number(7,2);
BEGIN
   if rabat_p > 50 then
      dbms_output.put_line('Nie mozna dać tak dużego rabatu ( '||rabat_p||'% )');
   else
      select cena_netto into cena_stara from p_produkt
      where nazwa_kategorii = kat;
      cena_nowa := cena_stara - (cena_stara * (rabat_p/100));
      update p_produkt
      set cena_netto = cena_nowa
      where nazwa_kategorii = kat;
      dbms_output.put_line('Przyznano rabat '||rabat_p||'% ');
   end if;
```

```
END RABAT;
/
--Wycofanie sprzedazy
create or replace PROCEDURE WYCOFANIE_SPRZEDAZY( kod in
p_sprzedaz.kod_produktu%type, faktura in p_sprzedaz.nr_faktury%type)
AS
BEGIN
   delete p_sprzedaz where kod_produktu = kod and nr_faktury = faktura;
   dbms_output.put_line('Wycofano sprzedaz produktu nr '||kod||' z faktury nr
'||faktura||'.');
END WYCOFANIE_SPRZEDAZY;
/

--TRIGGERY
--NAPRAWA:
create or replace TRIGGER P_TR_NAPRAWA
BEFORE INSERT OR UPDATE ON P_NAPRAWA
FOR EACH ROW
BEGIN
if INSERTING then
   :new.nr_naprawy := p_seq_nr_naprawy.nextval;
   :new.data_zakupu := ustaw_date(:new.nr_faktury);
   :new.data_odbioru := :new.data_zgloszenia+14;
end if;

if UPDATING('data_zgloszenia') then
   :new.data_odbioru := :new.data_zgloszenia+14;

end if;
```

```
END;
/
--PRODUKT
create or replace TRIGGER P_TR_PRODUKT
BEFORE INSERT OR UPDATE ON P_PRODUKT
FOR EACH ROW
BEGIN
if INSERTING then
   :new.kod_produktu := p_seq_kod_produktu.nextval;
   :new.cena_netto := licz_netto(:new.cena_zakupu);
end if;
if UPDATING('cena_zakupu') then
   :new.cena_netto := licz_netto(:new.cena_zakupu);
end if;

if UPDATING('cena_netto') then
   update p_sprzedaz
   set cena_brutto = licz_vat(:old.kod_produktu)
   where kod_produktu = :old.kod_produktu;
end if;
END;
/
--FAKTURA
create or replace TRIGGER P_TR_FAKTURA
BEFORE INSERT ON P_FAKTURA
FOR EACH ROW
BEGIN
   :new.nr_faktury := p_seq_nr_faktury.nextval;
END;
```

```
/
--KLIENT
create or replace TRIGGER P_TR_KLIENT
BEFORE INSERT ON P_KLIENT
FOR EACH ROW
BEGIN
   :new.nr_klienta := p_seq_nr_klienta.nextval;
END;
/
--SPRZEDAZ:
create or replace TRIGGER P_TR_SPRZEDAZ
BEFORE DELETE OR INSERT OR UPDATE ON P_Sprzedaz
FOR EACH ROW
BEGIN
if INSERTING then
   --produkt
   if czy_jest_w_magazynie(:new.kod_produktu, 0, :new.ilosc_zakupionych) >= 0 then
   update p_produkt
   set ilosc_w_magazynie = ilosc_w_magazynie -:new.ilosc_zakupionych
   where kod_produktu = :new.kod_produktu;
   --faktura
   :new.cena_brutto := licz_vat(:new.kod_produktu);

   update p_faktura
   set suma =suma + :new.cena_brutto * :new.ilosc_zakupionych
   where nr_faktury = :new.nr_faktury;
   else
   dbms_output.put_line('W magazynie nie ma wystarczajacej ilosci produktu');
   end if;
```

```
elsif DELETING then
   --produkt
   update p_produkt
   set ilosc_w_magazynie = ilosc_w_magazynie +:old.ilosc_zakupionych
   where kod_produktu = :old.kod_produktu;
   --faktura
   update p_faktura
   set suma =suma - :old.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;

end if;
if UPDATING ('kod_produktu') then
   --produkt
   update p_produkt
   set ilosc_w_magazynie=ilosc_w_magazynie+:new.ilosc_zakupionych
   where kod_produktu=:old.kod_produktu;

   update p_produkt
   set ilosc_w_magazynie=ilosc_w_magazynie-:new.ilosc_zakupionych
   where kod_produktu=:new.kod_produktu;

   --faktura
   update p_faktura
   set suma = suma - :old.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;

   :new.cena_brutto := licz_vat(:new.kod_produktu);

   update p_faktura
```

```
   set suma = suma + :new.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;

end if;
if UPDATING('ilosc_zakupionych') then
   --produkt
   if czy_jest_w_magazynie(:new.kod_produktu, :old.ilosc_zakupionych,
:new.ilosc_zakupionych) >= 0 then
   update p_produkt
   set ilosc_w_magazynie=ilosc_w_magazynie+:old.ilosc_zakupionych-
:new.ilosc_zakupionych
   where kod_produktu=:old.kod_produktu;

   --faktura
   update p_faktura
   set suma = suma - :old.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;

   update p_faktura
   set suma = suma + :old.cena_brutto * :new.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;
   else
   dbms_output.put_line('W magazynie nie ma wystarczajacej ilosci produktu.'|| chr(10) ||
'Ilosc zakupionych produktow nie zmienia sie');
   :new.ilosc_zakupionych := :old.ilosc_zakupionych;
   end if;

end if;

if UPDATING('nr_faktury') then
```

```
   --faktura
   update p_faktura
   set suma = suma - :old.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;

   update p_faktura
   set suma =suma + :old.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :new.nr_faktury;

end if;

if UPDATING('cena_brutto') then
   update p_faktura
   set suma =suma + :new.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;
end if;

END;
/

--PERSPEKTYWY

--Ilosc produktow sprzedanych w konkretnym miesiacu + zysk
create or replace view ile_sprzedano_miesiac as
select extract(month from data_zakupu) Miesiac, extract(year from data_zakupu) Rok,
sum(ilosc_zakupionych) Ilosc_sprzedanych_produktow,
sum(cena_brutto*ilosc_zakupionych) Suma_ze_sprzedazy,
sum(zysk(cena_netto,ilosc_zakupionych, cena_zakupu)) Zysk
from p_faktura f
join p_sprzedaz s on s.nr_faktury = f.nr_faktury
```

```
join p_produkt p on p.kod_produktu = s.kod_produktu
group by extract(year from data_zakupu),extract(month from data_zakupu)
order by Rok, Miesiac;

--Ilosc produktow sprzedanych w konkretnym roku
create or replace view ile_sprzedano_rok as
select extract(year from data_zakupu) Rok, sum(ilosc_zakupionych)
Ilosc_sprzedanych_produktow, sum(cena_brutto*ilosc_zakupionych) Suma_ze_sprzedazy,
sum(zysk(cena_netto,ilosc_zakupionych, cena_zakupu)) Zysk
from p_faktura f
join p_sprzedaz s on s.nr_faktury = f.nr_faktury
join p_produkt p on p.kod_produktu = s.kod_produktu
group by extract(year from data_zakupu)
order by Rok;

--Ilosc produktow sprzedanych w konkretnym dniu
create or replace view ile_sprzedano_dzien as
select data_zakupu Data, sum(ilosc_zakupionych) Ilosc_sprzedanych_produktow,
sum(cena_brutto*ilosc_zakupionych) Suma_ze_sprzedazy ,
sum(zysk(cena_netto,ilosc_zakupionych, cena_zakupu)) Zysk
from p_faktura f
join p_sprzedaz s on s.nr_faktury = f.nr_faktury
join p_produkt p on p.kod_produktu = s.kod_produktu
group by data_zakupu
order by data_zakupu;

--Ilosc produktow sprzedanych z konekretnej kategorii
create or replace view ile_sprzedano_kategoria as
```

```
select nazwa_kategorii Kategoria, sum(ilosc_zakupionych) Ilosc_sprzedanych_produktow,
sum(cena_brutto*ilosc_zakupionych) Suma_ze_sprzedazy,

sum(zysk(cena_netto,ilosc_zakupionych, cena_zakupu)) Zysk

from p_faktura f

join p_sprzedaz s on s.nr_faktury = f.nr_faktury

join p_produkt p on p.kod_produktu = s.kod_produktu

group by nazwa_kategorii

order by nazwa_kategorii;


--Ilosc sprzedanych produktow o danej nazwie

create or replace view ile_sprzedano_produkt as

select s.kod_produktu Kod, nazwa_produktu Nazwa_produktu, sum(ilosc_zakupionych)
Ilosc_sprzedanych_produktow, sum(cena_brutto*ilosc_zakupionych) Suma_ze_sprzedazy,

sum(zysk(cena_netto,ilosc_zakupionych, cena_zakupu)) Zysk

from p_produkt p

join p_sprzedaz s on p.kod_produktu = s.kod_produktu

group by s.kod_produktu, nazwa_produktu;


--Ilosc napraw danego produktu

create or replace view ile_napraw as

select n.kod_produktu Kod, nazwa_produktu Nazwa, count(nr_naprawy) IlOSC_NAPRAW

from p_naprawa n

join p_produkt p on p.kod_produktu = n.kod_produktu

group by n.kod_produktu, nazwa_produktu

order by count(nr_naprawy) desc;


--Faktury

create or replace view transakcje as

select f.nr_faktury, nazwa_firmy Firma, data_zakupu  Data_zakupu, imie_klient ||' '||
nazwisko_klient Klient,
```

```
nazwa_produktu as "Nazwa_produktu", ilosc_zakupionych Ilosc, cena_netto as
"cena_netto_za_szt", cena_brutto as "cena_brutto_za_szt"

from p_faktura f

join p_sprzedaz s on s.nr_faktury = f.nr_faktury

join p_produkt p on p.kod_produktu = s.kod_produktu

join p_klient k on k.nr_klienta = f.nr_klienta

order by f.nr_faktury;
```

5. Wprowadzanie danych

W pliku dodaj_dane_all.sql znajdują się zdania wprowadzające dane do wszystkich tabel w projekcie:

```
--Kategoria

SET DEFINE OFF


INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)

VALUES ('Procesor', 23);


INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)

VALUES ('Laptop', 23);


INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)

VALUES ('Smartfon', 23);


INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)

VALUES ('Płyta główna', 23);


INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)

VALUES ('RAM', 23);


INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)

VALUES ('Karta graficzna', 23);
```

```
INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)
VALUES ('Dysk twardy', 23);

INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)
VALUES ('Smartwatch', 23);

INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)
VALUES ('Kierownica', 23);

INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)
VALUES ('Słuchawki', 23);

INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)
VALUES ('Myszka', 23);

INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)
VALUES ('Monitor', 23);

INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)
VALUES ('Klawiatura', 23);

INSERT INTO P_KATEGORIA (NAZWA_KATEGORII, VAT)
VALUES ('Konsola', 23);

--Producent
SET DEFINE OFF

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT,
MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)
```

VALUES ('Intel', 'Aleje Jerozolimskie 146C', 'Warszawa', 801893052, 'intel@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('AMD', 'Grochowska 341', 'Warszawa', 177860121, 'amd@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Acer', 'Ireneusza Gugulskiego 1', 'Warszawa', 765995893, 'acer.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Lenovo', 'Daimlera 1', 'Warszawa', 866029867, 'lenovo.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Samsung', 'Postępu 14', 'Warszawa', 692799514, 'samsung.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Xiaomi', 'Jana Pawła II 82', 'Warszawa', 403777127, 'xiaomi.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Apple', 'Jana Pawła II 83', 'Warszawa', 442772294, 'apple.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('MSI', 'Ceramiczna 12', 'Marki', 248038006, 'msi.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Asus', 'Cybernetyki 9', 'Warszawa', 363063115, 'asus.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Corsair', 'Skłodowskiej-Curie 22', 'Katowice', 684082718, 'corsair.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('HyperX', 'Postępu 30', 'Warszawa', 748591429, 'hyperx.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Patriot', 'Jana Pawła II 84', 'Warszawa', 407682258, 'patriot.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('AFOX', 'Domaniewska 42A', 'Warszawa', 880182092, 'afox.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Gigabyte', 'Fabryczna 20', 'Wrocław', 212128186, 'gigabyte.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Seagate', 'Aleje Jerozolimskie 112C', 'Warszawa', 191141682, 'seagate.@gmail.com');

INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT, MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Huawei', 'Domaniewska 39A', 'Warszawa', 789155277, 'huawei.@gmail.com');

```
INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT,
MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Logitech', 'Aleje Jerozolimskie 181B', 'Warszawa', 459030454,
'logitech.@gmail.com');


INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT,
MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Accura', 'Dąbrówki 16', 'Katowice', 240247186, 'accura.@gmail.com');


INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT,
MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('JBL', 'Ostródzka 273/1', 'Warszawa', 318927817, 'jbl.@gmail.com');


INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT,
MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('Sony', 'Grzybowska 62', 'Warszawa', 308570668, 'sony.@gmail.com');


INSERT INTO P_PRODUCENT (NAZWA_PRODUCENTA, ADRES_PRODUCENT,
MIASTO_PRODUCENT, NR_TEL_PRODUCENT, MAIL_PRODUCENT)

VALUES ('California Access', 'Lipowa 184', 'Warszawa', 514323761, 'c.access@gmail.com');


--Pracownik

SET DEFINE OFF


INSERT INTO P_PRACOWNIK (NR_PRACOWNIKA, IMIE_PRACOWNIK,
NAZWISKO_PRACOWNIK, PLEC_PRACOWNIK, ADRES_PRACOWNIK,
KOD_POCZTOWY_PRACOWNIK, MIASTO_PRACOWNIK, NR_TEL_PRACOWNIK,
MAIL_PRACOWNIK)

VALUES (1, 'Andrzej', 'Stępniewski', 'M', 'Marszałkowska 18 m.3', '01-568', 'Warszawa',
561744102, 'a.stepniewski@onet.pl');


INSERT INTO P_PRACOWNIK (NR_PRACOWNIKA, IMIE_PRACOWNIK,
NAZWISKO_PRACOWNIK, PLEC_PRACOWNIK, ADRES_PRACOWNIK,
```

```
KOD_POCZTOWY_PRACOWNIK, MIASTO_PRACOWNIK, NR_TEL_PRACOWNIK,
MAIL_PRACOWNIK)

VALUES (2, 'Marcin', 'Miłowicz', 'M', 'Nowa 3', '00-602', 'Warszawa', 507371815,
'm.milowicz@gmail.com');


INSERT INTO P_PRACOWNIK (NR_PRACOWNIKA, IMIE_PRACOWNIK,
NAZWISKO_PRACOWNIK, PLEC_PRACOWNIK, ADRES_PRACOWNIK,
KOD_POCZTOWY_PRACOWNIK, MIASTO_PRACOWNIK, NR_TEL_PRACOWNIK,
MAIL_PRACOWNIK)

VALUES (3, 'Jan', 'Stefaniuk', 'M', 'Domańska 32/34 m.7', '03-769', 'Warszawa', 638710949,
'jan.stefaniuk@wp.pl');


INSERT INTO P_PRACOWNIK (NR_PRACOWNIKA, IMIE_PRACOWNIK,
NAZWISKO_PRACOWNIK, PLEC_PRACOWNIK, ADRES_PRACOWNIK,
KOD_POCZTOWY_PRACOWNIK, MIASTO_PRACOWNIK, NR_TEL_PRACOWNIK,
MAIL_PRACOWNIK)

VALUES (4, 'Martyna', 'Jantar', 'K', 'Biblioteczna 6', '04-111', 'Warszawa', 860411304,
'm.jantar@gmail.com');


--Powod naprawy

SET DEFINE OFF


INSERT INTO P_POWOD_NAPRAWY (KOD_POWODU, POWOD)

VALUES (1, 'uszkodzenia mechaniczne');


INSERT INTO P_POWOD_NAPRAWY (KOD_POWODU, POWOD)

VALUES (2, 'kontakt z cieczą/wilgocią');


INSERT INTO P_POWOD_NAPRAWY (KOD_POWODU, POWOD)

VALUES (3, 'zwarcie');


INSERT INTO P_POWOD_NAPRAWY (KOD_POWODU, POWOD)

VALUES (4, 'inne');
```

--Klient

SET DEFINE OFF

INSERT INTO P_KLIENT (IMIE_KLIENT, NAZWISKO_KLIENT, ADRES_KLIENT, KOD_POCZTOWY_KLIENT, MIASTO_KLIENT, NR_TEL_KLIENT, MAIL_KLIENT)

VALUES ('Jan', 'Kowalski', 'Jana Pawła II 14 m.3', '01-550', 'Warszawa', 961746101, 'j.kowalski@gmail.com');

INSERT INTO P_KLIENT (IMIE_KLIENT, NAZWISKO_KLIENT, ADRES_KLIENT, KOD_POCZTOWY_KLIENT, MIASTO_KLIENT, NR_TEL_KLIENT, MAIL_KLIENT)

VALUES ('Piotr', 'Nowak', 'Chylońska 137 m.90', '81-200', 'Gdynia', 567371855, 'p.nowak@wp.pl');

INSERT INTO P_KLIENT (IMIE_KLIENT, NAZWISKO_KLIENT, ADRES_KLIENT, KOD_POCZTOWY_KLIENT, MIASTO_KLIENT, NR_TEL_KLIENT, MAIL_KLIENT)

VALUES ('Beata', 'Wiśniewska', 'Morska 7', '81-780', 'Sopot', 338710749, 'beata.w@onet.pl');

INSERT INTO P_KLIENT (IMIE_KLIENT, NAZWISKO_KLIENT, ADRES_KLIENT, KOD_POCZTOWY_KLIENT, MIASTO_KLIENT, NR_TEL_KLIENT, MAIL_KLIENT)

VALUES ('Marta', 'Gałganowska', 'Warszawska 12', '19-304', 'Ełk', 862448304, 'm.galgan@gmail.com');

INSERT INTO P_KLIENT (IMIE_KLIENT, NAZWISKO_KLIENT, ADRES_KLIENT, KOD_POCZTOWY_KLIENT, MIASTO_KLIENT, NR_TEL_KLIENT, MAIL_KLIENT)

VALUES ('Anna', 'Fidusiewicz', 'Karolkowa 32 m.11', '01-610', 'Warszawa', 519447464, 'anna.fidu@onet.pl');

INSERT INTO P_KLIENT (IMIE_KLIENT, NAZWISKO_KLIENT, ADRES_KLIENT, KOD_POCZTOWY_KLIENT, MIASTO_KLIENT, NR_TEL_KLIENT, MAIL_KLIENT)

VALUES ('Tomasz', 'Akwinowicz', 'Graniczna 43', '05-120', 'Legionowo', 358448189, 'tomekzakwinu@gmail.com');

```
INSERT INTO P_KLIENT (IMIE_KLIENT, NAZWISKO_KLIENT, ADRES_KLIENT,
KOD_POCZTOWY_KLIENT, MIASTO_KLIENT, NR_TEL_KLIENT, MAIL_KLIENT)

VALUES ('Lucjan', 'Mostowiak', 'Wiejska 3', '05-110', 'Jabłonna', 954811718,
'lucek.most@gmail.com');


INSERT INTO P_KLIENT (IMIE_KLIENT, NAZWISKO_KLIENT, ADRES_KLIENT,
KOD_POCZTOWY_KLIENT, MIASTO_KLIENT, NR_TEL_KLIENT, MAIL_KLIENT)

VALUES ('Sławomir', 'Lewandowski', 'Wiejska 7 m.26', '01-302', 'Warszawa', 533165124,
'slawomir.l@onet.eu');


INSERT INTO P_KLIENT (IMIE_KLIENT, NAZWISKO_KLIENT, ADRES_KLIENT,
KOD_POCZTOWY_KLIENT, MIASTO_KLIENT, NR_TEL_KLIENT, MAIL_KLIENT)

VALUES ('Joanna', 'Kwoka', 'Łabędzia 12', '90-123', 'Łódź', 641937213, 'asia.kwoka@wp.pl');


INSERT INTO P_KLIENT (IMIE_KLIENT, NAZWISKO_KLIENT, ADRES_KLIENT,
KOD_POCZTOWY_KLIENT, MIASTO_KLIENT, NR_TEL_KLIENT, MAIL_KLIENT)

VALUES ('Janusz', 'Tracz', 'Kobiałka 131', '02-800', 'Warszawa', 589852147,
'janusz.tracz@op.pl');


--Produkt

SET DEFINE OFF


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('i5-10600', 'Intel', 'Procesor', 36, 238, 713.04);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('i7-10700', 'Intel', 'Procesor', 36, 265, 1250.61);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Ryzen 7 3700X', 'AMD', 'Procesor', 36, 261, 904.2);
```

```
INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Ryzen 5 3600X', 'AMD', 'Procesor', 36, 188, 1052.67);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Nitro 5', 'Acer', 'Laptop', 36, 170, 3109.93);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Legion 5-15ARH', 'Lenovo', 'Laptop', 24, 125, 2756.45);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Galaxy A42 5G', 'Samsung', 'Smartfon', 12, 68, 1095.08);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Mi 10T', 'Xiaomi', 'Smartfon', 12, 290, 1413.27);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('iPhone 12', 'Apple', 'Smartfon', 24, 199, 2968.53);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('MAG B460 TOMAHAWK', 'MSI', 'Płyta główna', 12, 123, 423.47);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('TUF GAMING H470-PRO', 'Asus', 'Płyta główna', 36, 239, 465.88);
```

```
INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Vengeance RGB Pro 16GB', 'Corsair', 'RAM', 60, 93, 352.77);

INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Fury Black 32GB', 'HyperX', 'RAM', 60, 75, 416.4);

INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Viper 4 8GB', 'Patriot', 'RAM', 60, 248, 134.25);

INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('GeForce RTX 3090 GAMING X TRIO 24G', 'MSI', 'Karta graficzna', 36, 12, 6214.21);

INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('GeForce GTX 1060 6GB', 'AFOX', 'Karta graficzna', 836, 63, 741.61);

INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Radeon RX 580 GAMING 8G', 'Gigabyte', 'Karta graficzna', 24, 140, 847.65);

INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('970 Evo Plus M.2 500GB', 'Samsung', 'Dysk twardy', 60, 235, 282.08);

INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('IronWolf 4TB', 'Seagate', 'Dysk twardy', 60, 178, 359.84);
```

```
INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Watch Fit czarny', 'Huawei', 'Smartwatch', 12, 269, 352.77);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Watch 5 GPS+Cellular', 'Apple', 'Smartwatch', 24, 102, 2473.67);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('G29', 'Logitech', 'Kierownica', 36, 299, 918.35);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Verde ACC S1719', 'Accura', 'Słuchawki', 12, 216, 49.42);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Tune 500 BT czarne', 'JBL', 'Słuchawki', 36, 168, 126.55);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('M235 czerwona', 'Logitech', 'Myszka', 12, 140, 69.99);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('M185 szara', 'Logitech', 'Myszka', 24, 77, 41.71);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('MX Master 3 grafitowa', 'Logitech', 'Myszka', 12, 168, 324.5);
```

```
INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('U28R550UQUX', 'Samsung', 'Monitor', 24, 85, 918.35);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('G413 Carbon', 'Logitech', 'Klawiatura', 36, 52, 253.8);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('PlayStation 5', 'Sony', 'Konsola', 12, 10, 1625.31);


INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA,
NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)

VALUES ('Shark CA-1418', 'California Access', 'Klawiatura', 12, 10, 142);


--Faktura

SET DEFINE OFF


INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)

VALUES (1, to_date('09.01.2021', 'DD.MM.YYYY'), 0);


INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)

VALUES (2, to_date('10.01.2021', 'DD.MM.YYYY'), 0);


INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)

VALUES (3, to_date('07.01.2021', 'DD.MM.YYYY'), 0);


INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)

VALUES (4, to_date('06.01.2021', 'DD.MM.YYYY'),0);
```

```
INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
VALUES (5, to_date('04.01.2021', 'DD.MM.YYYY'), 0);

INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
VALUES (6, to_date('05.01.2021', 'DD.MM.YYYY'), 0);

INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
VALUES (7, to_date('14.01.2021', 'DD.MM.YYYY'), 0);

INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
VALUES (8, to_date('10.01.2021', 'DD.MM.YYYY'), 0);

INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
VALUES (9, to_date('04.01.2021', 'DD.MM.YYYY'), 0);

INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
VALUES (10, to_date('05.01.2021', 'DD.MM.YYYY'), 0);

INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
VALUES (9, to_date('29.12.2020', 'DD.MM.YYYY'), 0);

INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
VALUES (3, to_date('05.10.2020', 'DD.MM.YYYY'), 0);

INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
VALUES (7, to_date('16.08.2020', 'DD.MM.YYYY'), 0);

INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
```

```
VALUES (2, to_date('21.01.2020', 'DD.MM.YYYY'), 0);

INSERT INTO P_FAKTURA (NR_KLIENTA, DATA_ZAKUPU, SUMA)
VALUES (6, to_date('21.08.2020', 'DD.MM.YYYY'), 0);

--Sprzedaz
SET DEFINE OFF

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (1, 1, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (2, 2, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (3, 3, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (4, 4, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (5, 1, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (6, 2, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (7, 3, 1);
```

```
INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (8, 10, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (9, 1, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (10, 3, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (11, 2, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (12, 4, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (13, 7, 3);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (14, 10, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (15, 9, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (16, 8, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (17, 1, 2);
```

```
INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (18, 2, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (19, 3, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (20, 7, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (21, 6, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (22, 6, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (23, 5, 3);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (24, 2, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (25, 3, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (26, 9, 3);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
```

```
VALUES (27, 7, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (28, 5, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (29, 4, 4);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (30, 3, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (31, 9, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (29, 7, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (20, 3, 4);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (13, 2, 3);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (13, 6, 1);
--
INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (20, 11, 1);
```

```
INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (26, 12, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (1, 13, 2);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (5, 14, 1);

INSERT INTO P_SPRZEDAZ (KOD_PRODUKTU, NR_FAKTURY, ILOSC_ZAKUPIONYCH)
VALUES (6, 15, 1);

--Naprawa
SET DEFINE OFF


INSERT INTO P_NAPRAWA (KOD_PRODUKTU, NR_SERYJNY, NR_FAKTURY, KOD_POWODU,
DATA_ZGLOSZENIA, NR_PRACOWNIKA)
VALUES (6, 'A934B018R5', 2, 1, '10/01/2021', 3);

INSERT INTO P_NAPRAWA (KOD_PRODUKTU, NR_SERYJNY, NR_FAKTURY, KOD_POWODU,
DATA_ZGLOSZENIA, NR_PRACOWNIKA)
VALUES (7, 'D934B048R528T13', 3, 1, '10/01/2021',  2);

INSERT INTO P_NAPRAWA (KOD_PRODUKTU, NR_SERYJNY, NR_FAKTURY, KOD_POWODU,
DATA_ZGLOSZENIA, NR_PRACOWNIKA)
values(9, 'A0453R86X8', 1, 2, '30/01/2021', 3);

INSERT INTO P_NAPRAWA (KOD_PRODUKTU, NR_SERYJNY, NR_FAKTURY, KOD_POWODU,
DATA_ZGLOSZENIA, NR_PRACOWNIKA)
values(10, 'B045FR86X8', 3, 1, '30/01/2021', 2);
```

```
INSERT INTO P_NAPRAWA (KOD_PRODUKTU, NR_SERYJNY, NR_FAKTURY, KOD_POWODU,
DATA_ZGLOSZENIA, NR_PRACOWNIKA)

values(27, 'C035FI86Y8', 7, 1, '21/01/2021', 3);


INSERT INTO P_NAPRAWA (KOD_PRODUKTU, NR_SERYJNY, NR_FAKTURY, KOD_POWODU,
DATA_ZGLOSZENIA, NR_PRACOWNIKA)

values(23, 'F035FI86H5', 5, 2, '23/01/2021', 1);


INSERT INTO P_NAPRAWA (KOD_PRODUKTU, NR_SERYJNY, NR_FAKTURY, KOD_POWODU,
DATA_ZGLOSZENIA, NR_PRACOWNIKA)

values(13, 'G245F086Z5', 2, 2, '31/01/2021', 4);
```

6. Usuwanie

Zdania umożliwiające usunięcie wszystkich elementów znajdują się w pliku projekt_drop.ddl:

```
DROP TABLE p_faktura CASCADE CONSTRAINTS;


DROP TABLE p_kategoria CASCADE CONSTRAINTS;


DROP TABLE p_klient CASCADE CONSTRAINTS;


DROP TABLE p_naprawa CASCADE CONSTRAINTS;


DROP TABLE p_powod_naprawy CASCADE CONSTRAINTS;


DROP TABLE p_pracownik CASCADE CONSTRAINTS;


DROP TABLE p_producent CASCADE CONSTRAINTS;


DROP TABLE p_produkt CASCADE CONSTRAINTS;
```

```
DROP TABLE p_sprzedaz CASCADE CONSTRAINTS;

DROP view ile_napraw;
DROP view ile_sprzedano_dzien;
DROP view ile_sprzedano_rok;
DROP view ile_sprzedano_miesiac;
DROP view ile_sprzedano_kategoria;
DROP view ile_sprzedano_produkt;
DROP view transakcje;

DROP procedure dostawa;
DROP procedure oddaj_do_naprawy;
DROP procedure rabat;
DROP procedure wycofanie_sprzedazy;

DROP function do_kiedy_gwarancja;
DROP function czy_gwarancja_wazna;
DROP function czy_jest_w_magazynie;
DROP function licz_netto;
DROP function licz_vat;
DROP function ustaw_date;
DROP function zysk;

DROP sequence p_seq_kod_produktu;
DROP sequence p_seq_nr_faktury;
DROP sequence p_seq_nr_klienta;
DROP sequence p_seq_nr_naprawy;
```

7. Instrukcja instalacji i deinstalacji

Na samym początku należy wykonać skrypt „projekt_create.ddl”. W tym pliku znajdują się zdania tworzące tabele, powiązania, perspektywy, wyzwalacze, funkcje, procedury, sekwencje, a także zdania ustawiające wartości domyślne. Następnie, aby wprowadzić dane do pliku, trzeba wykonać skrypt „dodaj_dane_all.sql”. Żeby usunąć wszystko należy wykonać skrypt „projekt_drop.ddl”.

Jeżeli pliki instalujące, deinstalujące oraz wprowadzające dane znajdują się na komputerze to wystarczy wpisać @ścieżka_do_pliku

Przykładowo:

@C:\Users\pc\Desktop\Studia\Semestr_3\BD\PROJEKT_BD\dane_do_projektu\projekt_create.ddl;

@C:\Users\pc\Desktop\Studia\Semestr_3\BD\PROJEKT_BD\dane_do_projektu\dodaj_dane_all.sql;

@C:\Users\pc\Desktop\Studia\Semestr_3\BD\PROJEKT_BD\dane_do_projektu\projekt_drop.ddl;

Po poprawnej instalacji w bazie powinny znajdować się takie produkty:

KOD_PRODUKTU	NAZWA_PRODUKTU	NAZWA_PRODUCENTA	NAZWA_KATEGORII	CENA_NETTO	GWARANCJA	ILOSC_W_MAGAZYNIE	CENA_ZAKUPU
1	i5-10600	Intel	Procesor	820	36	235	713,04
2	i7-10700	Intel	Procesor	1438,2	36	264	1250,61
3	Ryzen 7 3700X	AMD	Procesor	1039,83	36	260	904,2
4	Ryzen 5 3600X	AMD	Procesor	1210,57	36	187	1052,67
5	Nitro 5	Acer	Laptop	3576,42	36	167	3109,93
6	Legion 5-15ARH	Lenovo	Laptop	3169,92	24	122	2756,45
7	Galaxy A42 5G	Samsung	Smartfon	1259,34	12	67	1095,08
8	Mi 10T	Xiaomi	Smartfon	1625,26	12	288	1413,27
9	iPhone 12	Apple	Smartfon	3413,81	24	198	2968,53
10	MAG B460 TOMAHAWK	MSI	Płyta główna	486,99	12	122	423,47
11	TUF GAMING H470-PRO	Asus	Płyta główna	535,76	36	237	465,88
12	Vengeance RGB Pro 16GB	Corsair	RAM	405,69	60	92	352,77
13	Fury Black 32GB	HyperX	RAM	478,86	60	68	416,4
14	Viper 4 8GB	Patriot	RAM	154,39	60	246	134,25
15	GeForce RTX 3090 GAMING X TRIO 24G	MSI	Karta graficzna	7146,34	36	11	6214,21
16	GeForce GTX 1060 6GB	AFOX	Karta graficzna	852,85	836	61	741,61
17	Radeon RX 580 GAMING 8G	Gigabyte	Karta graficzna	974,8	24	138	847,65
18	970 Evo Plus M.2 500GB	Samsung	Dysk twardy	324,39	60	234	282,08
19	IronWolf 4TB	Seagate	Dysk twardy	413,82	60	177	359,84
20	Watch Fit czarny	Huawei	Smartwatch	405,69	12	262	352,77
21	Watch 5 GPS+Cellular	Apple	Smartwatch	2844,72	24	101	2473,67
22	G29	Logitech	Kierownica	1056,1	36	298	918,35
23	Verde ACC S1719	Accura	Słuchawki	56,83	12	213	49,42
24	Tune 500 BT czarne	JBL	Słuchawki	145,53	36	167	126,55
25	M235 czerwona	Logitech	Myszka	80,49	12	138	69,99
26	M185 szara	Logitech	Myszka	47,97	24	73	41,71
27	MX Master 3 grafitowa	Logitech	Myszka	373,18	12	166	324,5
28	U28R550UQUX	Samsung	Monitor	1056,1	24	84	918,35
29	G413 Carbon	Logitech	Klawiatura	291,87	36	47	253,8
30	PlayStation 5	Sony	Konsola	1869,11	12	9	1625,31
31	Shark CA-1418	California Access	Klawiatura	163,3	12	8	142

8. Zapewnienie poprawności danych

Podczas wprowadzania danych numer faktury, kod produktu, numer klienta oraz numer naprawy wprowadzają się automatycznie. Użycie sekwencji zapewnia unikatowość numerów, co umożliwia organizację.

```
CREATE SEQUENCE P_SEQ_KOD_PRODUKTU

minvalue 1

maxvalue 99999999999999999999

start with 1

increment by 1

cache 20;

--NR FAKTURY

CREATE SEQUENCE P_SEQ_NR_FAKTURY

minvalue 1

maxvalue 99999999999999999999

start with 1

increment by 1

cache 20;

--NR KLIENTA

CREATE SEQUENCE P_SEQ_NR_KLIENTA

minvalue 1

maxvalue 99999999999999999999

start with 1

increment by 1

cache 20;

--NR NAPRAWY

CREATE SEQUENCE P_SEQ_NR_NAPRAWY

minvalue 1

maxvalue 99999999999999999999

start with 1

increment by 1
```

cache 20;

KOD_PRODUKTU	NAZWA_PRODUKTU
1	i5-10600
2	i7-10700
3	Ryzen 7 3700X
4	Ryzen 5 3600X
5	Nitro 5
6	Legion 5-15ARH
7	Galaxy A42 5G

NR_FAKTURY	NR_KLIENTA	NAZWA_FIRMY	ADRES_FIRMA	KOD_POCZTOWY_FIRMA	MIASTO_FIRMA
1	1	revolution	Złota 18	01-908	Warszawa
2	2	revolution	Złota 18	01-908	Warszawa
3	3	revolution	Złota 18	01-908	Warszawa
4	4	revolution	Złota 18	01-908	Warszawa
5	5	revolution	Złota 18	01-908	Warszawa
6	6	revolution	Złota 18	01-908	Warszawa
7	7	revolution	Złota 18	01-908	Warszawa
8	8	revolution	Złota 18	01-908	Warszawa
9	9	revolution	Złota 18	01-908	Warszawa
10	10	revolution	Złota 18	01-908	Warszawa
11	9	revolution	Złota 18	01-908	Warszawa

NR_KLIENTA	IMIE_KLIENT	NAZWISKO_KLIENT	ADRES_KLIENT
1	Jan	Kowalski	Jana Pawła II 14 m.3
2	Piotr	Nowak	Chylońska 137 m.90
3	Beata	Wiśniewska	Morska 7
4	Marta	Gałganowska	Warszawska 12
5	Anna	Fidusiewicz	Karolkowa 32 m.11
6	Tomasz	Akwinowicz	Graniczna 43
7	Lucjan	Mostowiak	Wiejska 3
8	Sławomir	Lewandowski	Wiejska 7 m.26
9	Joanna	Kwoka	Łabędzia 12
10	Janusz	Tracz	Kobiałka 131

NR_NAPRAWY	KOD_PRODUKTU	NR_SERYJNY	NR_FAKTURY
1	6	A934B018R5	2
2	7	D934B048R528T13	3
3	9	A0453R86X8	1
4	10	B045FR86X8	3
5	27	C035FI86Y8	7
6	23	F035FI86H5	5
7	13	G245F086Z5	2

Podczas wprowadzania danych dotyczących sprzedaży niezbędna jest modyfikacja ilości sztuk danego produktu w magazynie. Jeśli klient kupił 3 produkty o kodzie 10, to ilość sztuk

tego produktu w magazynie zmniejsza się o 3. Jeśli jednak rozmyślił się i kupił 2 zamiast 3 sztuk, to jedna sztuka musi wrócić do magazynu. Klient nie może też kupić więcej sztuk produktu niż jest w magazynie. Automatycznie jest również obliczana cena netto produktu (na podstawie ceny za jaką sklep kupił produkt hurtowo), cena brutto suma oraz zysk na fakturze.

```
--Liczenie ceny brutto:

create or replace FUNCTION LICZ_VAT (kod p_sprzedaz.kod_produktu%type)

RETURN NUMBER AS

policzone number(7,2);

BEGIN

   select cena_netto+(cena_netto*vat/100) into policzone

   from p_produkt p

   join p_kategoria k on k.nazwa_kategorii = p.nazwa_kategorii

   where kod_produktu=kod;

   return policzone;

END LICZ_VAT;

/

--Liczenie ceny netto:

create or replace FUNCTION LICZ_NETTO(cena p_produkt.cena_zakupu%type)

RETURN NUMBER AS

wynik number(7,2);

BEGIN

   wynik := cena + cena*0.15;

   return wynik;

END LICZ_NETTO;

/

--Liczenie zysku:

create or replace FUNCTION ZYSK (cena p_produkt.cena_netto%type, ilosc
p_sprzedaz.ilosc_zakupionych%type, cena_z p_produkt.cena_zakupu%type)

RETURN NUMBER AS

zysk_s number(7,2);
```

```
BEGIN
   zysk_s := (cena - cena_z)*ilosc;
   return zysk_s;
END ZYSK;
    /
--PRODUKT
create or replace TRIGGER P_TR_PRODUKT
BEFORE INSERT OR UPDATE ON P_PRODUKT
FOR EACH ROW
BEGIN
if INSERTING then
   :new.kod_produktu := p_seq_kod_produktu.nextval;
   :new.cena_netto := licz_netto(:new.cena_zakupu);
end if;
if UPDATING('cena_zakupu') then
   :new.cena_netto := licz_netto(:new.cena_zakupu);
end if;

if UPDATING('cena_netto') then
   update p_sprzedaz
   set cena_brutto = licz_vat(:old.kod_produktu)
   where kod_produktu = :old.kod_produktu;
end if;
END;
/

--FAKTURA
create or replace TRIGGER P_TR_FAKTURA
BEFORE INSERT ON P_FAKTURA
```

```
FOR EACH ROW
BEGIN
   :new.nr_faktury := p_seq_nr_faktury.nextval;
END;
/
--KLIENT
create or replace TRIGGER P_TR_KLIENT
BEFORE INSERT ON P_KLIENT
FOR EACH ROW
BEGIN
   :new.nr_klienta := p_seq_nr_klienta.nextval;
END;
/
--SPRZEDAZ:
create or replace TRIGGER P_TR_SPRZEDAZ
BEFORE DELETE OR INSERT OR UPDATE ON P_Sprzedaz
FOR EACH ROW
BEGIN
if INSERTING then
   --produkt
   if czy_jest_w_magazynie(:new.kod_produktu, 0, :new.ilosc_zakupionych) >= 0 then
   update p_produkt
   set ilosc_w_magazynie = ilosc_w_magazynie -:new.ilosc_zakupionych
   where kod_produktu = :new.kod_produktu;
   --faktura
   :new.cena_brutto := licz_vat(:new.kod_produktu);

   update p_faktura
   set suma =suma + :new.cena_brutto * :new.ilosc_zakupionych
```

```
    where nr_faktury = :new.nr_faktury;
    else
    dbms_output.put_line('W magazynie nie ma wystarczajacej ilosci produktu');
    end if;
elsif DELETING then
    --produkt
    update p_produkt
    set ilosc_w_magazynie = ilosc_w_magazynie +:old.ilosc_zakupionych
    where kod_produktu = :old.kod_produktu;
    --faktura
    update p_faktura
    set suma =suma - :old.cena_brutto * :old.ilosc_zakupionych
    where nr_faktury = :old.nr_faktury;

end if;
if UPDATING ('kod_produktu') then
    --produkt
    update p_produkt
    set ilosc_w_magazynie=ilosc_w_magazynie+:new.ilosc_zakupionych
    where kod_produktu=:old.kod_produktu;

    update p_produkt
    set ilosc_w_magazynie=ilosc_w_magazynie-:new.ilosc_zakupionych
    where kod_produktu=:new.kod_produktu;

    --faktura
    update p_faktura
    set suma = suma - :old.cena_brutto * :old.ilosc_zakupionych
    where nr_faktury = :old.nr_faktury;
```

```
   :new.cena_brutto := licz_vat(:new.kod_produktu);


   update p_faktura
   set suma = suma + :new.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;


end if;
if UPDATING('ilosc_zakupionych') then
   --produkt
   if czy_jest_w_magazynie(:new.kod_produktu, :old.ilosc_zakupionych,
:new.ilosc_zakupionych) >= 0 then
   update p_produkt
   set ilosc_w_magazynie=ilosc_w_magazynie+:old.ilosc_zakupionych-
:new.ilosc_zakupionych
   where kod_produktu=:old.kod_produktu;


   --faktura
   update p_faktura
   set suma = suma - :old.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;


   update p_faktura
   set suma = suma + :old.cena_brutto * :new.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;
   else
   dbms_output.put_line('W magazynie nie ma wystarczajacej ilosci produktu.'|| chr(10) ||
'Ilosc zakupionych produktow nie zmienia sie');
   :new.ilosc_zakupionych := :old.ilosc_zakupionych;
   end if;
```

```
end if;

if UPDATING('nr_faktury') then
   --faktura
   update p_faktura
   set suma = suma - :old.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;

   update p_faktura
   set suma =suma + :old.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :new.nr_faktury;

end if;

if UPDATING('cena_brutto') then
   update p_faktura
   set suma =suma + :new.cena_brutto * :old.ilosc_zakupionych
   where nr_faktury = :old.nr_faktury;
end if;

END;
/
```

Po wykonaniu:

```
INSERT INTO P_PRODUKT (NAZWA_PRODUKTU, NAZWA_PRODUCENTA, NAZWA_KATEGORII, GWARANCJA, ILOSC_W_MAGAZYNIE, CENA_ZAKUPU)
VALUES ('i5-10600', 'Intel', 'Procesor', 36, 238, 713.04);
```

Cena netto ustawia się automatycznie:

KOD_PRODUKTU	NAZWA_PRODUKTU	NAZWA_PRODUCENTA	NAZWA_KATEGORII	CENA_NETTO	GWARANCJA	ILOSC_W_MAGAZYNIE	CENA_ZAKUPU
1	i5-10600	Intel	Procesor	820	36	235	713,04

Kiedy klient chce oddać produkt do naprawy, należy sprawdzić czy gwarancja już nie minęła. Jeśli tak, sklep nie dokona naprawy. Po oddaniu produktu do serwisu, klient musi otrzymać informacje za ile dni jego sprzęt będzie gotowy do odbioru (data ustawia się automatycznie, dwa tygodnie od daty oddania) oraz jaki pracownik dokonał naprawy. Należy również zapisać powód naprawy.

```
--Do kiedy jest ważna jest gwarancja:

create or replace FUNCTION DO_KIEDY_GWARANCJA (kod p_produkt.kod_produktu%type, data_z p_faktura.data_zakupu%type, faktura p_faktura.nr_faktury%type)

RETURN DATE AS

do_kiedy date;

BEGIN

   select Add_months(data_z,gwarancja) into do_kiedy

   from p_faktura f

   join p_sprzedaz s on s.nr_faktury = f.nr_faktury

   join p_produkt p on p.kod_produktu = s.kod_produktu

   where p.kod_produktu = kod and f.nr_faktury = faktura;

return do_kiedy;

END DO_KIEDY_GWARANCJA;

/

--Czy gwarancja jest jeszcze wazna:

create or replace FUNCTION CZY_GWARANCJA_WAZNA (kod p_produkt.kod_produktu%type, faktura p_faktura.nr_faktury%type, data_zgl p_naprawa.data_zgloszenia%type)

RETURN BOOLEAN AS

data_wygasniecia date;

data_zak date;

BEGIN


   select data_zakupu into data_zak

   from p_faktura f

   join p_sprzedaz s on f.nr_faktury = s.nr_faktury

   join p_produkt p on p.kod_produktu = s.kod_produktu
```

```
   where s.kod_produktu = kod and s.nr_faktury = faktura;

   data_wygasniecia := do_kiedy_gwarancja(kod, data_zak, faktura);

   if data_wygasniecia >= data_zgl then
   return true;
   else
   return false;
   end if;

END CZY_GWARANCJA_WAZNA;
/
create or replace TRIGGER P_TR_NAPRAWA
BEFORE INSERT OR UPDATE ON P_NAPRAWA
FOR EACH ROW
BEGIN
if INSERTING then
   :new.nr_naprawy := p_seq_nr_naprawy.nextval;
   :new.data_zakupu := ustaw_date(:new.nr_faktury);
   :new.data_odbioru := :new.data_zgloszenia+14;
end if;

if UPDATING('data_zgloszenia') then
   :new.data_odbioru := :new.data_zgloszenia+14;

end if;

END;
/
```

KOD_POWODU	POWOD
1	uszkodzenia mechaniczne
2	kontakt z cieczą/wilgocią
3	zwarcie
4	inne

NR_NAPRAWY	KOD_PRODUKTU	NR_SERYJNY	NR_FAKTURY	KOD_POWODU	DATA_ZAKUPU	DATA_ZGLOSZENIA	DATA_ODBIORU	NR_PRACOWNIKA
1	6	A934B018R5	2	1	10/01/2021	10/01/2021	24/01/2021	3

Można obniżyć cenę produktów wybranej kategorii:

```
--Rabat

create or replace PROCEDURE RABAT(kat p_kategoria.nazwa_kategorii%type, rabat_p
number)

AS

cena_stara number(7,2);

cena_nowa number(7,2);

BEGIN

  if rabat_p > 50 then

    dbms_output.put_line('Nie mozna dać tak dużego rabatu ( '||rabat_p||'% )');

  else

    select cena_netto into cena_stara from p_produkt

    where nazwa_kategorii = kat;

    cena_nowa := cena_stara - (cena_stara * (rabat_p/100));

    update p_produkt

    set cena_netto = cena_nowa

    where nazwa_kategorii = kat;

    dbms_output.put_line('Przyznano rabat '||rabat_p||'% ');

  end if;

END RABAT;

/
```

Zestawienia danych:

Naprawy produktów:

	KOD	NAZWA	ILOSC_NAPRAW
1	7	Galaxy A42 5G	1
2	10	MAG B460 TOMAHAWK	1
3	23	Verde ACC S1719	1
4	9	iPhone 12	1
5	6	Legion 5-15ARH	1
6	27	MX Master 3 grafitowa	1
7	13	Fury Black 32GB	1

Ilość sprzedanych produktów danego dnia:

	DATA	ILOSC_SPRZEDANYCH_PRODUKTOW	SUMA_ZE_SPRZEDAZY	ZYSK
1	21/01/2020	1	4399	466,49
2	16/08/2020	2	2017,2	213,92
3	21/08/2020	1	3899	413,47
4	05/10/2020	1	59	6,26
5	29/12/2020	1	499	52,92
6	04/01/2021	10	10877,42	1153,49
7	05/01/2021	7	9764,95	1035,52
8	06/01/2021	6	3424	363,1
9	07/01/2021	11	8428,99	893,87
10	09/01/2021	6	16403,59	1739,52
11	10/01/2021	12	15327,97	1625,44
12	14/01/2021	8	4042,02	428,65

Ilość sprzedanych produktów z podziałem na miesiące:

	MIESIAC	ROK	ILOSC_SPRZEDANYCH_PRODUKTOW	SUMA_ZE_SPRZEDAZY	ZYSK
1	1	2020	1	4399	466,49
2	8	2020	3	5916,2	627,39
3	10	2020	1	59	6,26
4	12	2020	1	499	52,92
5	1	2021	60	68268,94	7239,59

Ilość sprzedanych produktów z podziałem na lata:

	ROK	ILOSC_SPRZEDANYCH_PRODUKTOW	SUMA_ZE_SPRZEDAZY	ZYSK
1	2020	6	10873,2	1153,06
2	2021	60	68268,94	7239,59

Ile sztuk danego produktu zostało sprzedanych:

	KOD	NAZWA_PRODUKTU	ILOSC_SPRZEDANYCH_PRODUKTOW	SUMA_ZE_SPRZEDAZY	ZYSK
1	3	Ryzen 7 3700X	1	1278,99	135,63
2	7	Galaxy A42 5G	1	1548,99	164,26
3	10	MAG B460 TOMAHAWK	1	599	63,52
4	12	Vengeance RGB Pro 16GB	1	499	52,92
5	14	Viper 4 8GB	2	379,8	40,28
6	16	GeForce GTX 1060 6GB	2	2098,02	222,48
7	26	M185 szara	4	236	25,04
8	1	i5-10600	3	3025,8	320,88
9	22	G29	1	1299	137,75
10	24	Tune 500 BT czarne	1	179	18,98
11	30	PlayStation 5	1	2299,01	243,8
12	23	Verde ACC S1719	3	209,7	22,23
13	11	TUF GAMING H470-PRO	2	1317,96	139,76
14	13	Fury Black 32GB	7	4123	437,22
15	15	GeForce RTX 3090 GAMING X TRIO 24G	1	8790	932,13
16	17	Radeon RX 580 GAMING 8G	2	2398	254,3
17	25	M235 czerwona	2	198	21
18	31	Shark CA-1418	2	401,72	42,6
19	5	Nitro 5	3	13197	1399,47
20	20	Watch Fit czarny	7	3493	370,44
21	28	U28R550UQUX	1	1299	137,75
22	6	Legion 5-15ARH	3	11697	1240,41
23	8	Mi 10T	2	3998,14	423,98
24	19	IronWolf 4TB	1	509	53,98
25	27	MX Master 3 grafitowa	2	918,02	97,36
26	29	G413 Carbon	5	1795	190,35
27	4	Ryzen 5 3600X	1	1489	157,9
28	18	970 Evo Plus M.2 500GB	1	399	42,31
29	2	i7-10700	1	1768,99	187,59
30	9	iPhone 12	1	4198,99	445,28
31	21	Watch 5 GPS+Cellular	1	3499,01	371,05

Wszystkie transakcje:

	NR_FAKTURY	FIRMA	DATA_ZAKUPU	KLIENT	Nazwa_produktu	ILOSC	cena_netto_za_szt	cena_brutto_za_szt
1	1	revolution	09/01/2021	Jan Kowalski	i5-10600	1	820	1008,6
2	1	revolution	09/01/2021	Jan Kowalski	Nitro 5	2	3576,42	4399
3	1	revolution	09/01/2021	Jan Kowalski	iPhone 12	1	3413,81	4198,99
4	1	revolution	09/01/2021	Jan Kowalski	Radeon RX 580 GAMING 8G	2	974,8	1199
5	2	revolution	10/01/2021	Piotr Nowak	i7-10700	1	1438,2	1768,99
6	2	revolution	10/01/2021	Piotr Nowak	Legion 5-15ARH	2	3169,92	3899
7	2	revolution	10/01/2021	Piotr Nowak	TUF GAMING H470-PRO	2	535,76	658,98
8	2	revolution	10/01/2021	Piotr Nowak	Fury Black 32GB	3	478,86	589
9	2	revolution	10/01/2021	Piotr Nowak	970 Evo Plus M.2 500GB	1	324,39	399
10	2	revolution	10/01/2021	Piotr Nowak	Tune 500 BT czarne	1	145,53	179
11	3	revolution	07/01/2021	Beata Wiśniewska	Ryzen 7 3700X	1	1039,83	1278,99
12	3	revolution	07/01/2021	Beata Wiśniewska	Galaxy A42 5G	1	1259,34	1548,99
13	3	revolution	07/01/2021	Beata Wiśniewska	MAG B460 TOMAHAWK	1	486,99	599
14	3	revolution	07/01/2021	Beata Wiśniewska	IronWolf 4TB	1	413,82	509
15	3	revolution	07/01/2021	Beata Wiśniewska	Watch Fit czarny	4	405,69	499
16	3	revolution	07/01/2021	Beata Wiśniewska	M235 czerwona	2	80,49	99
17	3	revolution	07/01/2021	Beata Wiśniewska	PlayStation 5	1	1869,11	2299,01
18	4	revolution	06/01/2021	Marta Gałganowska	Ryzen 5 3600X	1	1210,57	1489
19	4	revolution	06/01/2021	Marta Gałganowska	Vengeance RGB Pro 16GB	1	405,69	499
20	4	revolution	06/01/2021	Marta Gałganowska	G413 Carbon	4	291,87	359
21	5	revolution	04/01/2021	Anna Fidusiewicz	Verde ACC S1719	3	56,83	69,9
22	5	revolution	04/01/2021	Anna Fidusiewicz	U28R550UQUX	1	1056,1	1299
23	6	revolution	05/01/2021	Tomasz Akwinowicz	Fury Black 32GB	1	478,86	589
24	6	revolution	05/01/2021	Tomasz Akwinowicz	Watch 5 GPS+Cellular	1	2844,72	3499,01
25	6	revolution	05/01/2021	Tomasz Akwinowicz	G29	1	1056,1	1299
26	7	revolution	14/01/2021	Lucjan Mostowiak	Fury Black 32GB	3	478,86	589
27	7	revolution	14/01/2021	Lucjan Mostowiak	Watch Fit czarny	2	405,69	499
28	7	revolution	14/01/2021	Lucjan Mostowiak	MX Master 3 grafitowa	2	373,18	459,01
29	7	revolution	14/01/2021	Lucjan Mostowiak	G413 Carbon	1	291,87	359
30	8	revolution	10/01/2021	Sławomir Lewandowski	GeForce GTX 1060 6GB	2	852,85	1049,01
31	9	revolution	04/01/2021	Joanna Kwoka	GeForce RTX 3090 GAMING X TRIO 24G	1	7146,34	8790
32	9	revolution	04/01/2021	Joanna Kwoka	M185 szara	3	47,97	59
33	9	revolution	04/01/2021	Joanna Kwoka	Shark CA-1418	2	163,3	200,86

10.Wyniki działania

revolution

FAKTURA
nr 1

Data wystawienia: 09/01/2021

Firma sprzedająca:

revolution
ul. Złota 18
01-908
Warszawa

Odbiorca:

Jan Kowalski
ul Jana Pawła II 14 m. 3
01-550
Warszawa

LP.	Nazwa produktu	Ilość	Cena netto za szt.	Cena brutto za szt.
1	i5-10600	1	820	1008.6
2	Nitro 5	2	3576.42	4399
3	iPhone 12	1	3413.81	4198.99
4	Radeon RX 580 GAMING	2	974.8	1199

Wartość faktury: 16403,59

revolution

Sprzedaż z podziałem na kategorie

Kategoria	Ilo	Suma	Zysk
Dysk twardy	2	908	96.29
Karta graficzna	5	13286.02	1408.91
Kierownica	1	1299	137.75
Klawiatura	7	2196.72	232.95
Konsola	1	2299.01	243.8
Laptop	6	24894	2639.88
Monitor	1	1299	137.75
Myszka	8	1352.02	143.4
Pyta gówna	3	1916.96	203.28
Procesor	6	7562.78	802
RAM	10	5001.8	530.42
Suchawki	4	388.7	41.21
Smartfon	4	9746.12	1033.52
Smartwatch	8	6992.01	741.49

Łączny zysk: 8392,65

revolution

NAPRAWY

Kod produktu	Nazwa produktu	Ilość napraw
6	Legion 5-15ARH	1
7	Galaxy A42 5G	1
9	iPhone 12	1
10	MAG B460 TOMAHAWK	1
13	Fury Black 32GB	1
23	Verde ACC S1719	1
27	MX Master 3 grafitowa	1

Łączna ilość napraw: 7

revolution

Sprzedaż w styczniu

Data sprzedaży	Ilość	Suma	Zysk
04.01.21 00:00	10	10877.42	1153.49
05.01.21 00:00	7	9764.95	1035.52
06.01.21 00:00	6	3424	363.1
07.01.21 00:00	11	8428.99	893.87
09.01.21 00:00	6	16403.59	1739.52
10.01.21 00:00	12	15327.97	1625.44
14.01.21 00:00	8	4042.02	428.65

Łączny zysk: 7239.59