

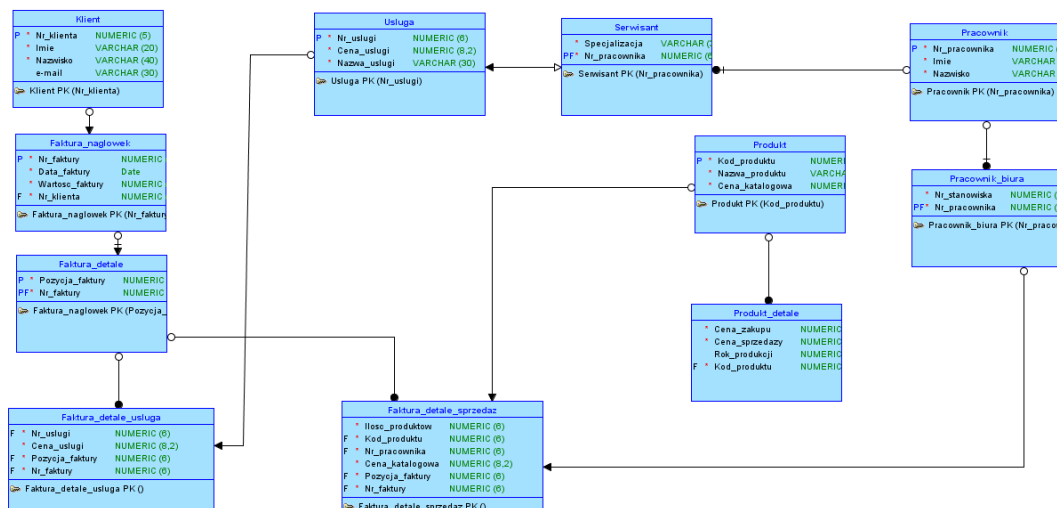
Analiza biznesowa projektowanej rzeczywistości :

System stworzony dla serwisu samochodowego zajmującego się wymianą opon, naprawami zawieszenia i silnika. Umożliwia sprzedaż produktów oraz wykonywanie usług. Dodatkowo pozwala przechowywać i modyfikować dane na temat klientów, faktur, usług, pracowników i sprzedawanych produktów.

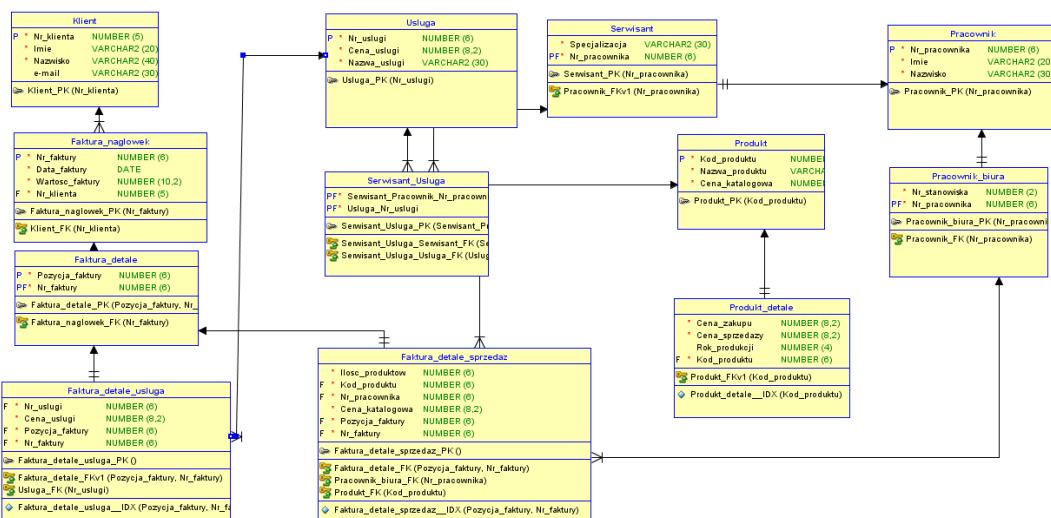
Tworzenie nowej faktury powinno odbywać się w następujący sposób:

1. Uzupełnienie danych klienta w przypadku gdy nie ma go jeszcze w bazie
2. Faktura nagłówek
3. Faktura_detale_usluga/Faktura_detale_sprzedaz

Model logiczny:



Model relacyjny :



Oprogramowanie tworzące bazę danych

--Tworzenie sekwencji

```
CREATE SEQUENCE seq_faktura START WITH 1 INCREMENT BY 1 MAXVALUE 9999999999999999 MINVALUE 1
NOCACHE;
```

```
CREATE SEQUENCE seq_klient START WITH 1 INCREMENT BY 1 MAXVALUE 9999999999 MINVALUE 1
NOCACHE;
```

```
CREATE SEQUENCE seq_poz_faktury START WITH 1 INCREMENT BY 1 MAXVALUE 99999999 MINVALUE 1
NOCACHE;
```

```
CREATE SEQUENCE seq_pracownik START WITH 1 INCREMENT BY 1 MAXVALUE 9999999999 MINVALUE 1
NOCACHE;
```

```
CREATE SEQUENCE seq_produkt START WITH 1 INCREMENT BY 1 MAXVALUE 9999999999999999 MINVALUE 1;
```

```
CREATE SEQUENCE seq_usługa START WITH 1 INCREMENT BY 1 MAXVALUE 9999999999 MINVALUE 1 NOCACHE;
```

--Tworzenie tabel

```
CREATE TABLE faktura_detale (
    pozycja_faktury NUMBER(6) NOT NULL,
    nr_faktury      NUMBER(6) NOT NULL
)
LOGGING;
```

```
ALTER TABLE faktura_detale ADD CONSTRAINT faktura_detale_pk PRIMARY KEY ( pozycja_faktury,
                                                                    nr_faktury );
```

```
CREATE TABLE faktura_detale_sprzedaz (
    ilosc_produktow  NUMBER(6) NOT NULL,
    kod_produktu     NUMBER(6) NOT NULL,
    nr_pracownika    NUMBER(6) NOT NULL,
    cena_katalogowa  NUMBER(8, 2) NOT NULL,
    pozycja_faktury  NUMBER(6) NOT NULL,
    nr_faktury       NUMBER(6) NOT NULL
)
LOGGING;
```

```
CREATE UNIQUE INDEX faktura_detale_sprzedaz__idx ON
    faktura_detale_sprzedaz (
        pozycja_faktury
    ASC,
        nr_faktury
    ASC )
```

LOGGING;

```
CREATE TABLE faktura_detale_usluga (  
    nr_uslugi    NUMBER(6) NOT NULL,  
    cena_uslugi  NUMBER(8, 2) NOT NULL,  
    pozycja_faktury NUMBER(6) NOT NULL,  
    nr_faktury   NUMBER(6) NOT NULL  
)  
LOGGING;
```

```
CREATE UNIQUE INDEX faktura_detale_usluga__idx ON  
    faktura_detale_usluga (  
        pozycja_faktury  
    ASC,  
        nr_faktury  
    ASC )  
LOGGING;
```

```
CREATE TABLE faktura_naglowek (  
    nr_faktury   NUMBER(6) NOT NULL,  
    data_faktury DATE NOT NULL,  
    wartosc_faktury NUMBER(10, 2) NOT NULL,  
    nr_klienta   NUMBER(5) NOT NULL  
)  
LOGGING;
```

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

```
CREATE TABLE klient (  
    nr_klienta  NUMBER(5) NOT NULL,  
    imie        VARCHAR2(20) NOT NULL,  
    nazwisko    VARCHAR2(40) NOT NULL,  
    "e-mail"    VARCHAR2(30)  
)  
LOGGING;
```

```
ALTER TABLE klient ADD CONSTRAINT klient_pk PRIMARY KEY ( nr_klienta );
```

```
CREATE TABLE pracownik (  
    nr_pracownika NUMBER(6) NOT NULL,  
    imie          VARCHAR2(20) NOT NULL,  
    nazwisko      VARCHAR2(30) NOT NULL  
)  
LOGGING;
```

```
ALTER TABLE pracownik ADD CONSTRAINT pracownik_pk PRIMARY KEY ( nr_pracownika );
```

```
CREATE TABLE pracownik_biura (  
    nr_stanowiska NUMBER(2) NOT NULL,  
    nr_pracownika NUMBER(6) NOT NULL  
)  
LOGGING;
```

```
ALTER TABLE pracownik_biura ADD CONSTRAINT pracownik_biura_pk PRIMARY KEY ( nr_pracownika );
```

```
CREATE TABLE produkt (  
    kod_produktu    NUMBER(6) NOT NULL,  
    nazwa_produktu  VARCHAR2(30) NOT NULL,  
    cena_katalogowa NUMBER(8, 2) NOT NULL  
)  
LOGGING;
```

```
ALTER TABLE produkt ADD CONSTRAINT produkt_pk PRIMARY KEY ( kod_produktu );
```

```
CREATE TABLE produkt_detale (  
    cena_zakupu    NUMBER(8, 2) NOT NULL,  
    cena_sprzedazy NUMBER(8, 2) NOT NULL,  
    rok_produkcji  NUMBER(4),  
    kod_produktu   NUMBER(6) NOT NULL  
)  
LOGGING;
```

```
CREATE UNIQUE INDEX produkt_detale__idx ON  
    produkt_detale (  
        kod_produktu  
    ASC )  
    LOGGING;
```

```
CREATE TABLE serwisant (  
    specjalizacja VARCHAR2(30) NOT NULL,  
    nr_pracownika  NUMBER(6) NOT NULL  
)  
LOGGING;
```

```
ALTER TABLE serwisant ADD CONSTRAINT serwisant_pk PRIMARY KEY ( nr_pracownika );
```

```
CREATE TABLE serwisant_usluga (  
    serwisant_pracownik_nr_pracownika NUMBER(6) NOT NULL,  
    usluga_nr_uslugi                   NUMBER(6) NOT NULL  
)  
LOGGING;
```

```
ALTER TABLE serwisant_usluga ADD CONSTRAINT serwisant_usluga_pk PRIMARY KEY (  
    serwisant_pracownik_nr_pracownika,  
    usluga_nr_uslugi );
```

```
CREATE TABLE usluga (  
    nr_uslugi    NUMBER(6) NOT NULL,  
    cena_uslugi  NUMBER(8, 2) NOT NULL,  
    nazwa_uslugi VARCHAR2(30) NOT NULL  
)  
LOGGING;
```

```
ALTER TABLE usluga ADD CONSTRAINT usluga_pk PRIMARY KEY ( nr_uslugi );
```

```
ALTER TABLE faktura_detale_sprzedaz
```

```

ADD CONSTRAINT faktura_detale_fk FOREIGN KEY ( pozycja_faktury,
                                             nr_faktury )
    REFERENCES faktura_detale ( pozycja_faktury,
                                nr_faktury )
NOT DEFERRABLE;

ALTER TABLE faktura_detale_uslug
ADD CONSTRAINT faktura_detale_fkv1 FOREIGN KEY ( pozycja_faktury,
                                             nr_faktury )
    REFERENCES faktura_detale ( pozycja_faktury,
                                nr_faktury )
NOT DEFERRABLE;

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

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

ALTER TABLE faktura_detale_sprzedaz
ADD CONSTRAINT pracownik_biura_fk FOREIGN KEY ( nr_pracownika )
    REFERENCES pracownik_biura ( nr_pracownika )
NOT DEFERRABLE;

ALTER TABLE pracownik_biura
ADD CONSTRAINT pracownik_fk FOREIGN KEY ( nr_pracownika )
    REFERENCES pracownik ( nr_pracownika )
NOT DEFERRABLE;

ALTER TABLE serwisant
ADD CONSTRAINT pracownik_fkv1 FOREIGN KEY ( nr_pracownika )
    REFERENCES pracownik ( nr_pracownika )
NOT DEFERRABLE;

ALTER TABLE faktura_detale_sprzedaz
ADD CONSTRAINT produkt_fk FOREIGN KEY ( kod_produkту )
    REFERENCES produkt ( kod_produkту )
NOT DEFERRABLE;

ALTER TABLE produkt_detale
ADD CONSTRAINT produkt_fkv1 FOREIGN KEY ( kod_produkту )
    REFERENCES produkt ( kod_produkту )
NOT DEFERRABLE;

ALTER TABLE serwisant_uslug
ADD CONSTRAINT serwisant_uslug_serwisant_fk FOREIGN KEY ( serwisant_pracownik_nr_pracownika )
    REFERENCES serwisant ( nr_pracownika )
NOT DEFERRABLE;

```

```

ALTER TABLE serwisant_usługa
  ADD CONSTRAINT serwisant_usługa_usługa_fk FOREIGN KEY ( usługa_nr_usługi )
    REFERENCES usługa ( nr_usługi )
  NOT DEFERRABLE;

```

```

ALTER TABLE faktura_detale_usługa
  ADD CONSTRAINT usługa_fk FOREIGN KEY ( nr_usługi )
    REFERENCES usługa ( nr_usługi )
  NOT DEFERRABLE;

```

--Tworzenie wyzwalaczy

```

CREATE OR REPLACE TRIGGER tr_ins_faktura_detale
  BEFORE INSERT ON Faktura_detale
  FOR EACH ROW
begin
:new.pozycja_faktury := seq_poz_faktury.nextval;
end;
/

```

```

CREATE OR REPLACE TRIGGER tr_ins_faktura_detale_sprzedaz
  BEFORE INSERT ON Faktura_detale_sprzedaz
  FOR EACH ROW
declare
v_cena_katalogowa faktura_detale_sprzedaz.cena_katalogowa%type;
v_pozycja_faktury faktura_detale_sprzedaz.pozycja_faktury%type;
begin
insert into faktura_detale(nr_faktury) values (:new.nr_faktury);
select cena_katalogowa into v_cena_katalogowa from produkt where kod_produktu = :new.kod_produktu;
select Max( pozycja_faktury) into v_pozycja_faktury from faktura_detale where nr_faktury =:new.nr_faktury;
:new.cena_katalogowa := v_cena_katalogowa;
:new.pozycja_faktury := v_pozycja_faktury ;

update faktura_naglowek
set wartosc_faktury = wartosc_faktury + (v_cena_katalogowa*:new.ilosc_produktow)
where nr_faktury =:new.nr_faktury;

end;
/

```

```

CREATE OR REPLACE TRIGGER tr_ins_faktura_detale_usługa
  BEFORE INSERT ON Faktura_detale_usługa
  FOR EACH ROW
declare
v_cena_usługi faktura_detale_usługa.cena_usługi%type;
v_pozycja_faktury faktura_detale_usługa.pozycja_faktury%type;
begin
insert into faktura_detale(nr_faktury) values (:new.nr_faktury);
select cena_usługi into v_cena_usługi from usługa where nr_usługi = :new.nr_usługi;
select Max( pozycja_faktury) into v_pozycja_faktury from faktura_detale where nr_faktury =:new.nr_faktury;
:new.cena_usługi := v_cena_usługi;
:new.pozycja_faktury := v_pozycja_faktury ;

```

```
update faktura_naglowek
set wartosc_faktury = wartosc_faktury + v_cena_uslugi
where nr_faktury =:new.nr_faktury;
```

```
end;
/
```

```
CREATE OR REPLACE TRIGGER tr_ins_klient
  BEFORE INSERT ON Klient
  FOR EACH ROW
begin
:new.nr_klienta := seq_klient.nextval;
end;
/
```

```
CREATE OR REPLACE TRIGGER tr_ins_pracownik
  BEFORE INSERT ON Pracownik
  FOR EACH ROW
begin
:new.nr_pracownika := seq_pracownik.nextval;
end;
/
```

```
CREATE OR REPLACE TRIGGER tr_ins_produkt
  BEFORE INSERT ON Produkt
  FOR EACH ROW
begin
:new.kod_produktu := seq_produkt.nextval;
end;
/
```

```
CREATE OR REPLACE TRIGGER tr_ins_usluga
  BEFORE INSERT ON Usluga
  FOR EACH ROW
begin
:new.nr_uslugi := seq_usluga.nextval;
end;
/
```

```
CREATE OR REPLACE TRIGGER "trigger tr_ins_faktura_naglowek "
  BEFORE INSERT ON Faktura_naglowek
  FOR EACH ROW
begin
:new.nr_faktury :=seq_faktura.nextval ;
:new.wartosc_faktury := 0 ;
:new.data_faktury :=sysdate;
end;
/
```

--Uzupełnianie danych:

--Klient:

```
insert into KLIENT(imie,nazwisko, "e-mail") values ('Jan', 'Kowalski', 'j.kowalski@op.pl');
```

```
insert into KLIENT(imie,nazwisko, "e-mail") values ('Antoni', 'Abacki', 'a.abacki@op.pl');
insert into KLIENT(imie,nazwisko, "e-mail") values ('Stanislaw', 'Patyk', 's.patyk@op.pl');
insert into KLIENT(imie,nazwisko, "e-mail") values ('Ferdynard', 'Bizon', 'f.bizon@op.pl');
insert into KLIENT(imie,nazwisko, "e-mail") values ('Anna', 'Nastula', 'a.nastula@op.pl');
```

--Pracownik:

```
insert into pracownik(imie,nazwisko) values ('Anna', 'Mirkowska');
insert into pracownik(imie,nazwisko) values ('Jan', 'Wacek');
insert into pracownik(imie,nazwisko) values ('Ryszard', 'Bak');
insert into pracownik(imie,nazwisko) values ('Tomasz', 'Zaglada');
insert into pracownik(imie,nazwisko) values ('Weronika', 'Ssak');
```

--Pracownik_biura:

```
insert into pracownik_biura(nr_pracownika, nr_stanowiska) values (1,1);
insert into pracownik_biura(nr_pracownika, nr_stanowiska) values (5,2);
```

--Serwisant:

```
insert into serwisant(nr_pracownika,specjalizacja) values (2,'opony');
insert into serwisant(nr_pracownika,specjalizacja) values (3,'zawieszenie');
insert into serwisant(nr_pracownika,specjalizacja) values (4,'silnik');
```

--Usługa:

```
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana opon', 120);
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana swiec', 20);
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana cewki', 15);
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana sprezyn zawieszenia', 100);
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana wahacza', 60);
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana lozyska', 80);
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana przegubu', 100);
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana drazka kierowniczego', 80);
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana amortyzatorow przod/ty', 150);
```

--Produkt:

```
insert into produkt(nazwa_produkту, cena_katalogowa) values ('Opona Dunlop 205/55/16','260');
insert into produkt(nazwa_produkту, cena_katalogowa) values ('Opona Nokian 205/55/16','298');
insert into produkt(nazwa_produkту, cena_katalogowa) values ('Opona Michelin 205/55/16','326');
insert into produkt(nazwa_produkту, cena_katalogowa) values ('Opona Hankook 205/55/16','213');
commit;
```

--Produkt detale:

```
insert into produkt_detale(cena_zakupu,cena_sprzedazy,rok_produkcji,kod_produkту) values
(200,260,2019,1);
insert into produkt_detale(cena_zakupu,cena_sprzedazy,rok_produkcji,kod_produkту) values
(230,298,2019,2);
insert into produkt_detale(cena_zakupu,cena_sprzedazy,rok_produkcji,kod_produkту) values
(270,326,2019,3);
insert into produkt_detale(cena_zakupu,cena_sprzedazy,rok_produkcji,kod_produkту) values
(150,213,2019,4);
```


--Widok drukujący fakturę

```
create or replace view FAKTURA as
select fn.nr_faktury,data_faktury,wartosc_faktury, kl.imie,nazwisko,fd.pozycja_faktury
,fds.ilosc_produktow,fds.cena_katalogowa,fdu.cena_uslugi
from klient kl
join faktura_naglowek fn on fn.nr_klienta = kl.nr_klienta
join faktura_detale fd on fd.nr_faktury = fn.nr_faktury
left outer join faktura_detale_sprzedaz fds on fds.pozycja_faktury = fd.pozycja_faktury
left outer join faktura_detale_usluga fdu on fdu.pozycja_faktury = fd.pozycja_faktury ;
```

--Widok drukujący raport zyskow z usług wykonanych przez pracownika serwisu

```
create or replace view zyski_uslugi_pracownik as
select se.nr_pracownika, pr.imie, pr.nazwisko, nvl(sum(fdu.cena_uslugi),0) "Przychod z uslug pracownika"
from uslugi us
join serwisant_usluga su on su.usluga_nr_uslugi = us.nr_uslugi
join serwisant se on se.nr_pracownika = su.serwisant_pracownik_nr_pracownika
join pracownik pr on pr.nr_pracownika = se.nr_pracownika
join faktura_detale_usluga fdu on fdu.nr_uslugi = us.nr_uslugi
group by se.nr_pracownika,pr.imie,pr.nazwisko, 'Przychod z uslug pracownika';
commit;
```

--Widok drukujący raport zysków ze sprzedaży produktów

```
create or replace view zyski_sprzedaz as
select pr.kod_produktu, pr.nazwa_produktu, pd.cena_zakupu*sum(fds.ilosc_produktow) "Lacny koszt
zakupu", pd.cena_sprzedazy*sum(fds.ilosc_produktow) "Lacny koszt sprzedazy",
pd.cena_sprzedazy*sum(fds.ilosc_produktow)-pd.cena_zakupu*sum(fds.ilosc_produktow) "Lacny zysk z
produktu"
from produkt pr
join produkt_detale pd on pd.kod_produktu = pr.kod_produktu
join faktura_detale_sprzedaz fds on fds.kod_produktu= pr.kod_produktu
group by pr.kod_produktu,pr.nazwa_produktu,pd.cena_sprzedazy, fds.ilosc_produktow,pd.cena_zakupu;
```

--Widok drukujący raport zysków ze sprzedaży produktów wraz z danymi pracownika dokonującego sprzedaży

```
create or replace view zyski_sprzedaz_pracownik as
select pr.imie,pr.nazwisko, pd.cena_zakupu*sum(fds.ilosc_produktow) "Lacny koszt zakupu",
pd.cena_sprzedazy*sum(fds.ilosc_produktow) "Lacny koszt sprzedazy",
pd.cena_sprzedazy*sum(fds.ilosc_produktow)-pd.cena_zakupu*sum(fds.ilosc_produktow) as zysk
from pracownik pr
join pracownik_biura prb on prb.nr_pracownika =pr.nr_pracownika
join faktura_detale_sprzedaz fds on fds.nr_pracownika = prb.nr_pracownika
join produkt pro on pro.kod_produktu = fds.kod_produktu
join produkt_detale pd on pd.kod_produktu = pro.kod_produktu
group by pr.nr_pracownika,pr.imie,pr.nazwisko,pd.cena_sprzedazy, fds.ilosc_produktow,pd.cena_zakupu;
```

--Procedura odpowiedzialna za dodawanie klientów do bazy

```
create or replace procedure pr_insert_nowy_klient
(v_imie klient.imie%type, v_nazwisko klient.nazwisko%type, v_email klient."e-mail"%type)
as
begin
insert into klient(imie,nazwisko,"e-mail") values (v_imie,v_nazwisko,v_email);
commit;
end;
```

-- Wywołanie

```
declare v_imie klient.imie%type;
v_nazwisko klient.nazwisko%type;
v_email klient."e-mail"%type;
begin
v_imie := :Imie ;
v_nazwisko := :Nazwisko ;
v_email := :email ;
pr_insert_nowy_klient(v_imie,v_nazwisko,v_email);
end;
```

--Procedura odpowiedzialna za aktualizowanie danych o kliencie

```
create or replace procedure pr_update_klient
(v_nr_klienta klient.nr_klienta%type, v_imie klient.imie%type, v_nazwisko klient.nazwisko%type, v_email
klient."e-mail"%type)
as
begin
update klient
set imie = v_imie,
nazwisko = v_nazwisko,
"e-mail" = v_email
where nr_klienta = v_nr_klienta ;
commit;
end;
```

-- Wywołanie

```
declare v_nr_klienta klient.nr_klienta%type;
v_imie klient.imie%type;
v_nazwisko klient.nazwisko%type;
v_email klient."e-mail"%type;
begin
v_nr_klienta := :Nr_klienta ;
v_imie := :Imie ;
v_nazwisko := :Nazwisko ;
v_email := :email ;
pr_update_klient(v_nr_klienta,v_imie,v_nazwisko,v_email);
end;
```

--Procedura odpowiedzialna za usuwanie klientow z bazy

```
create or replace procedure pr_delete_klient
(v_nr_klienta klient.nr_klienta%type)
as
begin
```

```

delete klient
where nr_klienta = v_nr_klienta ;
commit;
end;

```

--Wywołanie

```

declare v_nr_klienta klient.nr_klienta%type ;
begin
v_nr_klienta := :Nr_klienta_do_usuniecia;
pr_delete_klient(v_nr_klienta);
end;

```

--Funkcja oblicza liczbę pozycji na danej fakturze :

```

create or replace function fn_ile_poz_na_fakturze(v_nr_faktury number)
return number
as
v_ilosc_poz number;
begin
select count(*) into v_ilosc_poz from faktura_detale
where nr_faktury = v_nr_faktury
group by nr_faktury;
return v_ilosc_poz;
end fn_ile_poz_na_fakturze;

```

--Wywołanie

```

select nr_faktury, fn_ile_poz_na_fakturze(nr_faktury) from faktura_detale
group by nr_faktury;

```

	NR_FAKTURY	FN_ILE_POZ_NA_FAKTURZE(NR_FAKTURY)
1	1	6
2	2	1

--Funkcja oblicza ilość wydanych pieniędzy przez klientów naszej działalności.

```

create or replace function fn_ile_wydane(v_nr_klienta number)
return number
as
v_ile_wydane number;
begin
select sum(wartosc_faktury) into v_ile_wydane from faktura_naglowek
where nr_klienta = v_nr_klienta
group by nr_klienta;
return v_ile_wydane;
end fn_ile_wydane ;

```

```

select nr_klienta,fn_ile_wydane(nr_klienta) from faktura_naglowek
group by nr_klienta;

```

	NR_KLIENTA	FN_ILE_WYDANE(NR_KLIENTA)
1	1	3532
2	2	120

--Funkcja oblicza średnią ilość wydanych pieniędzy przez danego klienta

```
create or replace function fn_sr_ile_wydane(v_nr_klienta number)
return number
as
v_sr_ile_wydane number;
begin
select avg(wartosc_faktury) into v_sr_ile_wydane from faktura_naglowek
where nr_klienta = v_nr_klienta
group by nr_klienta;
return v_sr_ile_wydane ;
end fn_sr_ile_wydane;
```

```
select nr_klienta,fn_sr_ile_wydane(nr_klienta) from faktura_naglowek
group by nr_klienta
```

	NR_KLIENTA	FN_SR_ILE_WYDANE(NR_KLIENTA)
1	1	1766
2	2	120

Skrypt:

DROP TABLE faktura_detale CASCADE CONSTRAINTS;

DROP TABLE faktura_detale_sprzedaz CASCADE CONSTRAINTS;

DROP TABLE faktura_detale_usluga CASCADE CONSTRAINTS;

DROP TABLE faktura_naglowek CASCADE CONSTRAINTS;

DROP TABLE klient CASCADE CONSTRAINTS;

DROP TABLE pracownik CASCADE CONSTRAINTS;

DROP TABLE pracownik_biura CASCADE CONSTRAINTS;

DROP TABLE produkt CASCADE CONSTRAINTS;

DROP TABLE produkt_detale CASCADE CONSTRAINTS;

DROP TABLE serwisant CASCADE CONSTRAINTS;

DROP TABLE serwisant_usluga CASCADE CONSTRAINTS;

DROP TABLE usluga CASCADE CONSTRAINTS;

DROP SEQUENCE seq_faktura;

DROP SEQUENCE seq_klient;

DROP SEQUENCE seq_poz_faktury;

DROP SEQUENCE seq_pracownik;

DROP SEQUENCE seq_produk;

```
DROP SEQUENCE seq_usluga;

DROP FUNCTION fn_ile_poz_na_fakturze;

DROP FUNCTION fn_ile_wydane;

DROP FUNCTION fn_sr_ile_wydane;

DROP PROCEDURE pr_delete_klient;

DROP PROCEDURE pr_insert_nowy_klient;

DROP PROCEDURE pr_update_klient;

DROP VIEW FAKTURA;

DROP VIEW ZYSKI_SPRZEDAZ;

DROP VIEW ZYSKI_SPRZEDAZ_PRACOWNIK;

DROP VIEW ZYSKI_USLUGI_PRACOWNIK;
```

```
CREATE SEQUENCE seq_faktura START WITH 1 INCREMENT BY 1 MAXVALUE 9999999999999999 MINVALUE 1 NOCACHE;
```

```
CREATE SEQUENCE seq_klient START WITH 1 INCREMENT BY 1 MAXVALUE 9999999999 MINVALUE 1 NOCACHE;
```

```
CREATE SEQUENCE seq_poz_faktury START WITH 1 INCREMENT BY 1 MAXVALUE 99999999 MINVALUE 1 NOCACHE;
```

```
CREATE SEQUENCE seq_pracownik START WITH 1 INCREMENT BY 1 MAXVALUE 9999999999 MINVALUE 1 NOCACHE;
```

```
CREATE SEQUENCE seq_produkty START WITH 1 INCREMENT BY 1 MAXVALUE 9999999999999999 MINVALUE 1;
```

```
CREATE SEQUENCE seq_usluga START WITH 1 INCREMENT BY 1 MAXVALUE 9999999999 MINVALUE 1 NOCACHE;
```

```
CREATE TABLE faktura_detale (
  pozycja_faktury NUMBER(6) NOT NULL,
  nr_faktury      NUMBER(6) NOT NULL
)
LOGGING;
```

```
ALTER TABLE faktura_detale ADD CONSTRAINT faktura_detale_pk PRIMARY KEY ( pozycja_faktury,
                                                                    nr_faktury );
```

```
CREATE TABLE faktura_detale_sprzedaz (
  ilosc_produktoz NUMBER(6) NOT NULL,
  kod_produktoz   NUMBER(6) NOT NULL,
  nr_pracownika   NUMBER(6) NOT NULL,
  cena_katalogowa NUMBER(8, 2) NOT NULL,
  pozycja_faktury NUMBER(6) NOT NULL,
  nr_faktury      NUMBER(6) NOT NULL
)
LOGGING;
```

```
CREATE UNIQUE INDEX faktura_detale_sprzedaz_idx ON
  faktura_detale_sprzedaz (
    pozycja_faktury
  )
ASC,
```

```

        nr_faktury
    ASC )
    LOGGING;

CREATE TABLE faktura_detale_usluga (
    nr_uslugi    NUMBER(6) NOT NULL,
    cena_uslugi  NUMBER(8, 2) NOT NULL,
    pozycja_faktury  NUMBER(6) NOT NULL,
    nr_faktury   NUMBER(6) NOT NULL
)
LOGGING;

CREATE UNIQUE INDEX faktura_detale_usluga__idx ON
    faktura_detale_usluga (
        pozycja_faktury
    ASC,
        nr_faktury
    ASC )
    LOGGING;

CREATE TABLE faktura_naglowek (
    nr_faktury   NUMBER(6) NOT NULL,
    data_faktury DATE NOT NULL,
    wartosc_faktury  NUMBER(10, 2) NOT NULL,
    nr_klienta    NUMBER(5) NOT NULL
)
LOGGING;

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

CREATE TABLE klient (
    nr_klienta  NUMBER(5) NOT NULL,
    imie        VARCHAR2(20) NOT NULL,
    nazwisko    VARCHAR2(40) NOT NULL,
    "e-mail"    VARCHAR2(30)
)
LOGGING;

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

CREATE TABLE pracownik (
    nr_pracownika  NUMBER(6) NOT NULL,
    imie           VARCHAR2(20) NOT NULL,
    nazwisko       VARCHAR2(30) NOT NULL
)
LOGGING;

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

CREATE TABLE pracownik_biura (
    nr_stanowiska  NUMBER(2) NOT NULL,
    nr_pracownika  NUMBER(6) NOT NULL
)
LOGGING;

ALTER TABLE pracownik_biura ADD CONSTRAINT pracownik_biura_pk PRIMARY KEY ( nr_pracownika );

CREATE TABLE produkt (
    kod_produkту  NUMBER(6) NOT NULL,
    nazwa_produkту  VARCHAR2(30) NOT NULL,
    cena_katalogowa  NUMBER(8, 2) NOT NULL
)

```

LOGGING;

ALTER TABLE produkt ADD CONSTRAINT produkt_pk PRIMARY KEY (kod_produktu);

```
CREATE TABLE produkt_detale (
  cena_zakupu   NUMBER(8, 2) NOT NULL,
  cena_sprzedazy NUMBER(8, 2) NOT NULL,
  rok_produkcji NUMBER(4),
  kod_produktu  NUMBER(6) NOT NULL
)
```

LOGGING;

```
CREATE UNIQUE INDEX produkt_detale__idx ON
  produkt_detale (
    kod_produktu
  )
  ASC )
  LOGGING;
```

```
CREATE TABLE serwisant (
  specjalizacja VARCHAR2(30) NOT NULL,
  nr_pracownika NUMBER(6) NOT NULL
)
```

LOGGING;

ALTER TABLE serwisant ADD CONSTRAINT serwisant_pk PRIMARY KEY (nr_pracownika);

```
CREATE TABLE serwisant_usluga (
  serwisant_pracownik_nr_pracownika NUMBER(6) NOT NULL,
  usluga_nr_uslugi                   NUMBER(6) NOT NULL
)
```

LOGGING;

ALTER TABLE serwisant_usluga ADD CONSTRAINT serwisant_usluga_pk PRIMARY KEY (serwisant_pracownik_nr_pracownika, usluga_nr_uslugi);

```
CREATE TABLE usluga (
  nr_uslugi   NUMBER(6) NOT NULL,
  cena_uslugi NUMBER(8, 2) NOT NULL,
  nazwa_uslugi VARCHAR2(30) NOT NULL
)
```

LOGGING;

ALTER TABLE usluga ADD CONSTRAINT usluga_pk PRIMARY KEY (nr_uslugi);

```
ALTER TABLE faktura_detale_sprzedaz
  ADD CONSTRAINT faktura_detale_fk FOREIGN KEY ( pozycja_faktury,
                                                    nr_faktury )
    REFERENCES faktura_detale ( pozycja_faktury,
                                nr_faktury )
  NOT DEFERRABLE;
```

```
ALTER TABLE faktura_detale_usluga
  ADD CONSTRAINT faktura_detale_fkv1 FOREIGN KEY ( pozycja_faktury,
                                                    nr_faktury )
    REFERENCES faktura_detale ( pozycja_faktury,
                                nr_faktury )
  NOT DEFERRABLE;
```

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

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

```
ALTER TABLE faktura_detale_sprzedaz
ADD CONSTRAINT pracownik_biura_fk FOREIGN KEY ( nr_pracownika )
REFERENCES pracownik_biura ( nr_pracownika )
NOT DEFERRABLE;
```

```
ALTER TABLE pracownik_biura
ADD CONSTRAINT pracownik_fk FOREIGN KEY ( nr_pracownika )
REFERENCES pracownik ( nr_pracownika )
NOT DEFERRABLE;
```

```
ALTER TABLE serwisant
ADD CONSTRAINT pracownik_fkv1 FOREIGN KEY ( nr_pracownika )
REFERENCES pracownik ( nr_pracownika )
NOT DEFERRABLE;
```

```
ALTER TABLE faktura_detale_sprzedaz
ADD CONSTRAINT produkt_fk FOREIGN KEY ( kod_produktu )
REFERENCES produkt ( kod_produktu )
NOT DEFERRABLE;
```

```
ALTER TABLE produkt_detale
ADD CONSTRAINT produkt_fkv1 FOREIGN KEY ( kod_produktu )
REFERENCES produkt ( kod_produktu )
NOT DEFERRABLE;
```

```
ALTER TABLE serwisant_usluga
ADD CONSTRAINT serwisant_usluga_serwisant_fk FOREIGN KEY ( serwisant_pracownik_nr_pracownika )
REFERENCES serwisant ( nr_pracownika )
NOT DEFERRABLE;
```

```
ALTER TABLE serwisant_usluga
ADD CONSTRAINT serwisant_usluga_usluga_fk FOREIGN KEY ( usluga_nr_uslugi )
REFERENCES usluga ( nr_uslugi )
NOT DEFERRABLE;
```

```
ALTER TABLE faktura_detale_usluga
ADD CONSTRAINT usluga_fk FOREIGN KEY ( nr_uslugi )
REFERENCES usluga ( nr_uslugi )
NOT DEFERRABLE;
```

```
CREATE OR REPLACE TRIGGER tr_ins_faktura_detale
BEFORE INSERT ON Faktura_detale
FOR EACH ROW
begin
:new.pozycja_faktury := seq_poz_faktury.nextval;
end;
/
```

```
CREATE OR REPLACE TRIGGER tr_ins_faktura_detale_sprzedaz
BEFORE INSERT ON Faktura_detale_sprzedaz
FOR EACH ROW
declare
v_cena_katalogowa faktura_detale_sprzedaz.cena_katalogowa%type;
v_pozycja_faktury faktura_detale_sprzedaz.pozycja_faktury%type;
begin
insert into faktura_detale(nr_faktury) values (:new.nr_faktury);
```



```

select cena_katalogowa into v_cena_katalogowa from produkt where kod_produktu = :new.kod_produktu;
select Max( pozycja_faktury) into v_pozycja_faktury from faktura_detale where nr_faktury =:new.nr_faktury;
:new.cena_katalogowa := v_cena_katalogowa;
:new.pozycja_faktury := v_pozycja_faktury ;

```

```

update faktura_naglowek
set wartosc_faktury = wartosc_faktury + (v_cena_katalogowa*:new.ilosc_produktow)
where nr_faktury =:new.nr_faktury;

```

```
end;
```

```
/
```

```

CREATE OR REPLACE TRIGGER tr_ins_faktura_detale_usluga
  BEFORE INSERT ON Faktura_detale_usluga
  FOR EACH ROW

```

```
declare
```

```
v_cena_uslugi faktura_detale_usluga.cena_uslugi%type;
```

```
v_pozycja_faktury faktura_detale_usluga.pozycja_faktury%type;
```

```
begin
```

```
insert into faktura_detale(nr_faktury) values (:new.nr_faktury);
```

```
select cena_uslugi into v_cena_uslugi from usluga where nr_uslugi = :new.nr_uslugi;
```

```
select Max( pozycja_faktury) into v_pozycja_faktury from faktura_detale where nr_faktury =:new.nr_faktury;
```

```
:new.cena_uslugi := v_cena_uslugi;
```

```
:new.pozycja_faktury := v_pozycja_faktury ;
```

```
update faktura_naglowek
```

```
set wartosc_faktury = wartosc_faktury + v_cena_uslugi
```

```
where nr_faktury =:new.nr_faktury;
```

```
end;
```

```
/
```

```

CREATE OR REPLACE TRIGGER tr_ins_klient
  BEFORE INSERT ON Klient
  FOR EACH ROW

```

```
begin
```

```
:new.nr_klienta := seq_klient.nextval;
```

```
end;
```

```
/
```

```

CREATE OR REPLACE TRIGGER tr_ins_pracownik
  BEFORE INSERT ON Pracownik
  FOR EACH ROW

```

```
begin
```

```
:new.nr_pracownika := seq_pracownik.nextval;
```

```
end;
```

```
/
```

```

CREATE OR REPLACE TRIGGER tr_ins_produkt
  BEFORE INSERT ON Produkt
  FOR EACH ROW

```

```
begin
```

```
:new.kod_produktu := seq_produkt.nextval;
```

```
end;
```

```
/
```

```

CREATE OR REPLACE TRIGGER tr_ins_usluga
  BEFORE INSERT ON Usluga
  FOR EACH ROW

```

```
begin
```

```
:new.nr_uslugi := seq_usluga.nextval;
```

```
end;
```

/

```
CREATE OR REPLACE TRIGGER "trigger tr_ins_faktura_naglowek "
```

```
BEFORE INSERT ON Faktura_naglowek
```

```
FOR EACH ROW
```

```
begin
```

```
:new.nr_faktury :=seq_faktura.nextval ;
```

```
:new.wartosc_faktury := 0 ;
```

```
:new.data_faktury :=sysdate;
```

```
end;
```

/

```
insert into KLIENT(imie,nazwisko, "e-mail") values ('Jan', 'Kowalski', 'j.kowalski@op.pl');
```

```
insert into KLIENT(imie,nazwisko, "e-mail") values ('Antoni', 'Abacki', 'a.abacki@op.pl');
```

```
insert into KLIENT(imie,nazwisko, "e-mail") values ('Stanislaw', 'Patyk', 's.patyk@op.pl');
```

```
insert into KLIENT(imie,nazwisko, "e-mail") values ('Ferdynard', 'Bizon', 'f.bizon@op.pl');
```

```
insert into KLIENT(imie,nazwisko, "e-mail") values ('Anna', 'Nastula', 'a.nastula@op.pl');
```

```
insert into pracownik(imie,nazwisko) values ('Anna', 'Mirkowska');
```

```
insert into pracownik(imie,nazwisko) values ('Jan', 'Wacek');
```

```
insert into pracownik(imie,nazwisko) values ('Ryszard', 'Bak');
```

```
insert into pracownik(imie,nazwisko) values ('Tomasz', 'Zaglada');
```

```
insert into pracownik(imie,nazwisko) values ('Weronika', 'Ssak');
```

```
insert into pracownik_biura(nr_pracownika, nr_stanowiska) values (1,1);
```

```
insert into pracownik_biura(nr_pracownika, nr_stanowiska) values (5,2);
```

```
insert into serwisant(nr_pracownika,specjalizacja) values (2,'opony');
```

```
insert into serwisant(nr_pracownika,specjalizacja) values (3,'zawieszenie');
```

```
insert into serwisant(nr_pracownika,specjalizacja) values (4,'silnik');
```

```
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana opon', 120);
```

```
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana swiec', 20);
```

```
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana cewki', 15);
```

```
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana sprzyn zawieszenia', 100);
```

```
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana wahacza', 60);
```

```
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana lozyska', 80);
```

```
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana przegubu', 100);
```

```
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana drazka kierowniczego', 80);
```

```
insert into uslugu(nazwa_uslugi, cena_uslugi) values ('wymiana amortyzatorow przod/ty', 150);
```

```
insert into produkt(nazwa_produkту, cena_katalogowa) values ('Opona Dunlop 205/55/16','260');
```

```
insert into produkt(nazwa_produkту, cena_katalogowa) values ('Opona Nokian 205/55/16','298');
```

```
insert into produkt(nazwa_produkту, cena_katalogowa) values ('Opona Michelin 205/55/16','326');
```

```
insert into produkt(nazwa_produkту, cena_katalogowa) values ('Opona Hankook 205/55/16','213');
```

```
commit;
```

```
insert into produkt_detale(cena_zakupu,cena_sprzedazy,rok_produkци,kod_produkту) values (200,260,2019,1);
```

```
insert into produkt_detale(cena_zakupu,cena_sprzedazy,rok_produkци,kod_produkту) values (230,298,2019,2);
```

```
insert into produkt_detale(cena_zakupu,cena_sprzedazy,rok_produkци,kod_produkту) values (270,326,2019,3);
```

```
insert into produkt_detale(cena_zakupu,cena_sprzedazy,rok_produkци,kod_produkту) values (150,213,2019,4);
```

```
select fn.nr_faktury,data_faktury,wartosc_faktury, kl.imie,nazwisko,fd.pozycja_faktury
```

```
,fds.ilosc_produkту,fds.cena_katalogowa,fdu.cena_uslugi
```

```
from klient kl
```

```
join faktura_naglowek fn on fn.nr_klienta = kl.nr_klienta
```

```
join faktura_detale fd on fd.nr_faktury = fn.nr_faktury
```

```
left outer join faktura_detale_sprzedaz fds on fds.pozycja_faktury = fd.pozycja_faktury
```

```
left outer join faktura_detale_uslugu fdu on fdu.pozycja_faktury = fd.pozycja_faktury ;
```

```
create or replace view zyski_uslugi_pracownik as
```

```
select se.nr_pracownika, pr.imie, pr.nazwisko, nvl(sum(fdu.cena_uslugi),0) "Przychod z uslug pracownika"
```

```
from uslugu us
```

```
join serwisant_uslugu su on su.uslugu_nr_uslugi = us.nr_uslugi
```

```
join serwisant se on se.nr_pracownika = su.serwisant_pracownik_nr_pracownika
```

```

join pracownik pr on pr.nr_pracownika = se.nr_pracownika
join faktura_detale_usluga fdu on fdu.nr_uslugi = us.nr_uslugi
group by se.nr_pracownika,pr.imie,pr.nazwisko, 'Przychod z uslug pracownika';
commit;

```

```

create or replace view zyski_sprzedaz as
select pr.kod_produkту, pr.nazwa_produkту, pd.cena_zakupu*sum(fds.ilosc_produkту) "Laczny koszt zakupu",
pd.cena_sprzedazy*sum(fds.ilosc_produkту) "Laczny koszt sprzedazy",
pd.cena_sprzedazy*sum(fds.ilosc_produkту)-pd.cena_zakupu*sum(fds.ilosc_produkту) "Laczny zysk z produkту"
from produkt pr
join produkt_detale pd on pd.kod_produkту = pr.kod_produkту
join faktura_detale_sprzedaz fds on fds.kod_produkту= pr.kod_produkту
group by pr.kod_produkту,pr.nazwa_produkту,pd.cena_sprzedazy, fds.ilosc_produkту,pd.cena_zakupu;

```

```

create or replace view zyski_sprzedaz_pracownik as
select pr.imie,pr.nazwisko, pd.cena_zakupu*sum(fds.ilosc_produkту) "Laczny koszt zakupu",
pd.cena_sprzedazy*sum(fds.ilosc_produkту) "Laczny koszt sprzedazy",
pd.cena_sprzedazy*sum(fds.ilosc_produkту)-pd.cena_zakupu*sum(fds.ilosc_produkту) as zysk
from pracownik pr
join pracownik_biura prb on prb.nr_pracownika =pr.nr_pracownika
join faktura_detale_sprzedaz fds on fds.nr_pracownika = prb.nr_pracownika
join produkt pro on pro.kod_produkту = fds.kod_produkту
join produkt_detale pd on pd.kod_produkту = pro.kod_produkту
group by pr.nr_pracownika,pr.imie,pr.nazwisko,pd.cena_sprzedazy, fds.ilosc_produkту,pd.cena_zakupu;
create or replace procedure pr_insert_nowy_klient
(v_imie klient.imie%type, v_nazwisko klient.nazwisko%type, v_email klient."e-mail"%type)
as
begin
insert into klient(imie,nazwisko,"e-mail") values (v_imie,v_nazwisko,v_email);
commit;
end;

```

```

create or replace procedure pr_update_klient
(v_nr_klienta klient.nr_klienta%type, v_imie klient.imie%type, v_nazwisko klient.nazwisko%type, v_email klient."e-mail"%type)
as
begin
update klient
set imie = v_imie,
nazwisko = v_nazwisko,
"e-mail"= v_email
where nr_klienta = v_nr_klienta ;
commit;
end;
create or replace procedure pr_delete_klient
(v_nr_klienta klient.nr_klienta%type)
as
begin
delete klient
where nr_klienta = v_nr_klienta ;
commit;
end;

```

```

create or replace function fn_ile_poz_na_fakturze(v_nr_faktury number)
return number
as
v_ilosc_poz number;
begin
select count(*) into v_ilosc_poz from faktura_detale
where nr_faktury = v_nr_faktury
group by nr_faktury;
return v_ilosc_poz;
end fn_ile_poz_na_fakturze;

```

```

create or replace function fn_ile_wydane(v_nr_klienta number)
return number
as

```

```

v_ile_wydane number;
begin
select sum(wartosc_faktury) into v_ile_wydane from faktura_naglowek
where nr_klienta = v_nr_klienta
group by nr_klienta;
return v_ile_wydane;
end fn_ile_wydane ;

create or replace function fn_sr_ile_wydane(v_nr_klienta number)
return number
as
v_sr_ile_wydane number;
begin
select avg(wartosc_faktury) into v_sr_ile_wydane from faktura_naglowek
where nr_klienta = v_nr_klienta
group by nr_klienta;
return v_sr_ile_wydane ;
end fn_sr_ile_wydane;

```

Instalacja:

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

Raporty:



28/02/2020

FAKTURA

NR_FAKTU	DATA_FAKTURY	WARTOSC_FAKTURY	IMIE	NAZWISKO	POZYCJA_FAKTURY	ILOSC_PRODUKTOW	CENA_KATALOGOWA	CENA_USLUGI
1	28.02.20 10:59	3532	Jan	Kowalski	2	null	null	120
1	28.02.20 10:59	3532	Jan	Kowalski	3	null	null	20
2	28.02.20 12:05	120	Antoni	Abacki	5	null	null	120
1	28.02.20 10:59	3532	Jan	Kowalski	6	null	null	120
1	28.02.20 10:59	3532	Jan	Kowalski	7	4	298	null
1	28.02.20 10:59	3532	Jan	Kowalski	8	4	260	null
1	28.02.20 10:59	3532	Jan	Kowalski	4	4	260	null



28/02/2020

Zysk sprzedaży produktu

KOD_PRODUKTU	NAZWA_PRODUKTU	Lacny koszt zakupu	Lacny koszt sprzedazy	Lacny zysk z produktu
1	Opona Dunlop	1600	2080	480
2	Opona Nokian	920	1192	272



28/02/2020

Zysk z usług uzyskany przez pracownika serwisu

NR_PRACOWNIKA	IMIE	NAZWISKO	Przychod z usług
2	Jan	Wacek	360
4	Tomasz	Zaglada	20



28/02/2020

Zestawienie zyskow ze sprzedazy pracownika

IMIE	NAZWISKO	Laczny koszt zakupu	Laczny koszt sprzedazy	ZYSK
Anna	Mirkowska	920	1192	272
Anna	Mirkowska	1600	2080	480

