

Dokumentacja projektu bazy danych

Prowadzący: mgr inż. Józef Woźniak

Wykonała: Sylwia Pytko

Grupa: I4Z6S1

Nr albumu: 59947

Data: 30.01.16r

1.	Analiza biznesowa opracowywanej rzeczywistości	4
2.	Model logiczny i relacyjny bazy danych	4
3.	Oprogramowanie tworzące bazę danych oraz generator danych	5
1.	Stworzenie tabel wraz z ich atrybutami	5
2.	Dodanie relacji między tabelami	7
3.	Utworzenie sekwencji i przypisanie wartości defaultowych.....	8
4.	Wprowadzenie danych do tabel.....	9
1.	Klient.....	9
2.	Kolekcja	13
3.	Rodzaj	15
4.	Rozmiar.....	17
5.	Model	20
6.	Materiał	26
7.	Produkt – wprowadzanie procedurą.....	29
5.	<i>Generator faktury.....</i>	<i>33</i>
	<i>// 1. sekwencja tworząca klucze główne dla faktur</i>	<i>33</i>
	<i>// 2. funkcja losująca klienta z przedziału ich id.....</i>	<i>33</i>
	<i>// 3. funkcja losująca produkt z przedziału ich id</i>	<i>33</i>
	<i>// 4. funkcja pobierająca cenę wybranego produktu</i>	<i>34</i>
	<i>// 5. wyzwalacz na insert</i>	<i>34</i>
	<i>// 6. funkcja tworząca pozycję:</i>	<i>34</i>
	<i>// 7. funkcja tworząca nową fakturę:.....</i>	<i>35</i>
6.	Perspektywy klasyczne	36
1.	Perspektywa koniec_gwarancji	36
2.	Perspektywa model_o_rozmiarze.....	36
7.	Perspektywy zmaterializowane	37
1.	Perspektywa zmaterializowana m_wybrany_material_przed_30	37
2.	Perspektywa zmaterializowana m_wybrany_material	38
3.	Perspektywa zmaterializowana m_zarobek_kolekcji.....	38
4.	Perspektywa zmaterializowana m_ile_zakupów_klient	39
4.	Skrypty wdrożeniowe instalujące i deinstalujące projekt	39
5.	Instrukcja instalacji projektu i sprawdzenia jej poprawności.....	40
1.	Instrukcja instalacji	40

2.	Sprawdzenie poprawności.....	41
6.	Wyniki działania perspektyw analitycznych w postaci zrzutów ekranowych	42

1. Analiza biznesowa opracowywanej rzeczywistości

Temat: Zakład/sklep tapicerski

Opis:

Sklep sprzedaje meble obite materiałem. W sklepie występują kolekcje stworzone w pewnym roku przez pewnego projektanta. Kolekcja posiada wiele modeli.

Model należy do rodzaju które mają przeznaczenie i pewną gwarancję.

Model posiada też swój rozmiar ale dany rozmiar może mieć wiele modeli.

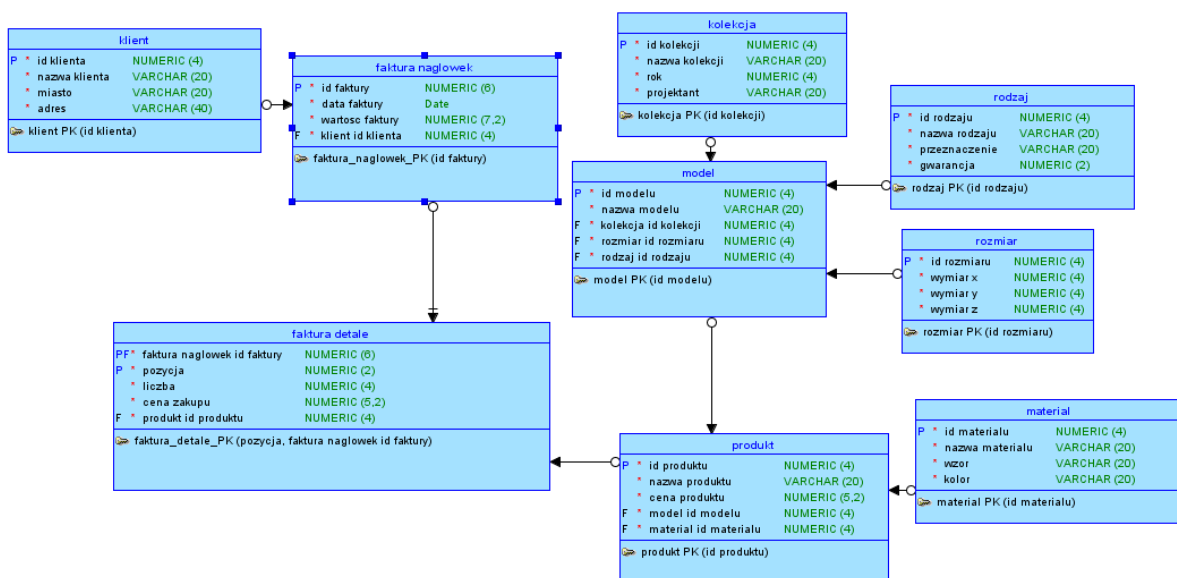
Sklep oferuje wszystkie dostępne modele we wszystkich posiadanych materiałach.

Klient kupuje produkty w systemie faktury wielopozycyjnej. Może za jedną fakturą kupić wiele różnych produktów w różnych ilościach. Informacje o kliencie nie są dla sklepu istotne. Są potrzebne tylko do ewentualnego „wydrukowania” faktury.

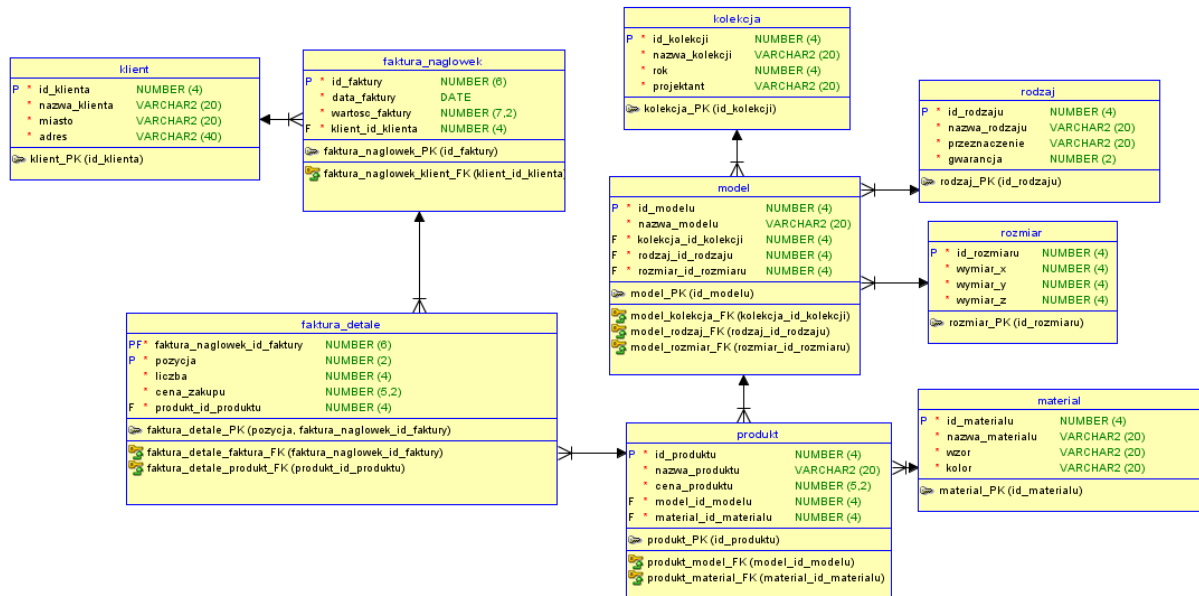
W zestawieniu danych sprzedarzy bardziej interesujący jest jednak fakt sprzedarzy konkretnych rodzaj mebli lub też modeli czy użytych materiałów niż zakup konkretnego modelu w konkretnym obiciu (produkcie) gdyż produktów jest bardzo duża ilość.

2. Model logiczny i relacyjny bazy danych

Model logiczny



Model relacyjny



3. Oprogramowanie tworzące bazę danych oraz generator danych

1. Stworzenie tabel wraz z ich atrybutami

CREATE TABLE faktura_detale

```
(
    faktura_naglowek_id_faktury NUMERIC (6) NOT NULL ,
    pozycja                NUMERIC (2) NOT NULL ,
    liczba                 NUMERIC (4) NOT NULL ,
    cena_zakupu            NUMERIC (5,2) NOT NULL ,
    produkt_id_produkту    NUMERIC (4) NOT NULL
);
```

ALTER TABLE faktura_detale ADD CONSTRAINT faktura_detale_PK PRIMARY KEY (pozycja, faktura_naglowek_id_faktury);

CREATE TABLE faktura_naglowek

```
(
    id_faktury    NUMERIC (6) NOT NULL ,
```

```

data_faktury    DATE NOT NULL ,
wartosc_faktury NUMERIC (7,2) NOT NULL ,
klient_id_klienta NUMERIC (4) NOT NULL
);
ALTER TABLE faktura_naglowek ADD CONSTRAINT faktura_naglowek_PK PRIMARY KEY ( id_faktury );

```

```

CREATE TABLE klient
(
id_klienta    NUMERIC (4) NOT NULL ,
nazwa_klienta VARCHAR (20) NOT NULL ,
miasto       VARCHAR (20) NOT NULL ,
adres        VARCHAR (40) NOT NULL
);
ALTER TABLE klient ADD CONSTRAINT klient_PK PRIMARY KEY ( id_klienta );

```

```

CREATE TABLE kolekcja
(
id_kolekcji    NUMERIC (4) NOT NULL ,
nazwa_kolekcji VARCHAR (20) NOT NULL ,
rok            NUMERIC (4) NOT NULL ,
projektant     VARCHAR (20) NOT NULL
);
ALTER TABLE kolekcja ADD CONSTRAINT kolekcja_PK PRIMARY KEY ( id_kolekcji );

```

```

CREATE TABLE material
(
id_materialu    NUMERIC (4) NOT NULL ,
nazwa_materialu VARCHAR (20) NOT NULL ,
wzor           VARCHAR (20) NOT NULL ,
kolor          VARCHAR (20) NOT NULL
);
ALTER TABLE material ADD CONSTRAINT material_PK PRIMARY KEY ( id_materialu );

```

```

CREATE TABLE model
(
id_modelu      NUMERIC (4) NOT NULL ,
nazwa_modelu   VARCHAR (20) NOT NULL ,
kolekcja_id_kolekcji NUMERIC (4) NOT NULL ,
rodzaj_id_rodzaju NUMERIC (4) NOT NULL ,
rozmiar_id_rozmiaru NUMERIC (4) NOT NULL
);

```

```
ALTER TABLE model ADD CONSTRAINT model_PK PRIMARY KEY ( id_modelu );
```

```
CREATE TABLE produkt
```

```
(  
  id_produktu      NUMERIC (4) NOT NULL ,  
  nazwa_produktu   VARCHAR (20) NOT NULL ,  
  cena_produktu    NUMERIC (5,2) NOT NULL ,  
  model_id_modelu   NUMERIC (4) NOT NULL ,  
  material_id_materialu NUMERIC (4) NOT NULL  
);
```

```
ALTER TABLE produkt ADD CONSTRAINT produkt_PK PRIMARY KEY ( id_produktu );
```

```
CREATE TABLE rodzaj
```

```
(  
  id_rodzaju  NUMERIC (4) NOT NULL ,  
  nazwa_rodzaju VARCHAR (20) NOT NULL ,  
  przeznaczenie VARCHAR (20) NOT NULL ,  
  gwarancja   NUMERIC (2) NOT NULL  
);
```

```
ALTER TABLE rodzaj ADD CONSTRAINT rodzaj_PK PRIMARY KEY ( id_rodzaju );
```

```
CREATE TABLE rozmiar
```

```
(  
  id_rozmiaru NUMERIC (4) NOT NULL ,  
  wymiar_x    NUMERIC (4) NOT NULL ,  
  wymiar_y    NUMERIC (4) NOT NULL ,  
  wymiar_z    NUMERIC (4) NOT NULL  
);
```

```
ALTER TABLE rozmiar ADD CONSTRAINT rozmiar_PK PRIMARY KEY ( id_rozmiaru );
```

2. Dodanie relacji między tabelami

```
ALTER TABLE faktura_detale ADD CONSTRAINT faktura_detale_faktura_FK FOREIGN KEY (  
faktura_naglowek_id_faktury ) REFERENCES faktura_naglowek ( id_faktury );
```

```
ALTER TABLE faktura_detale ADD CONSTRAINT faktura_detale_produkt_FK FOREIGN KEY (  
produkt_id_produktu ) REFERENCES produkt ( id_produktu );
```

```
ALTER TABLE faktura_naglowek ADD CONSTRAINT faktura_naglowek_klient_FK FOREIGN KEY (  
klient_id_klienta ) REFERENCES klient ( id_klienta );
```

```
ALTER TABLE model ADD CONSTRAINT model_kolekcja_FK FOREIGN KEY ( kolekcja_id_kolekcji )  
REFERENCES kolekcja ( id_kolekcji ) ;
```

```
ALTER TABLE model ADD CONSTRAINT model_rodzaj_FK FOREIGN KEY ( rodzaj_id_rodzaju )  
REFERENCES rodzaj ( id_rodzaju ) ;
```

```
ALTER TABLE model ADD CONSTRAINT model_rozmiar_FK FOREIGN KEY ( rozmiar_id_rozmiaru )  
REFERENCES rozmiar ( id_rozmiaru ) ;
```

```
ALTER TABLE produkt ADD CONSTRAINT produkt_material_FK FOREIGN KEY ( material_id_materialu )  
REFERENCES material ( id_materialu ) ;
```

```
ALTER TABLE produkt ADD CONSTRAINT produkt_model_FK FOREIGN KEY ( model_id_modelu )  
REFERENCES model ( id_modelu ) ;
```

3. Utworzenie sekwencji i przypisanie wartości defaultowych

```
alter table material
```

```
  modify wzor DEFAULT 'Gładki'; //Większość materiałów nie będzie miała wzoru
```

```
alter table rodzaj
```

```
  modify gwarancja DEFAULT 24; //większość rodzajów będzie miała 2 letnią gwarancję
```

//Dla każdej tabeli (oprócz fakturowych) tworze sekwencję liczącą klucz główny

```
CREATE SEQUENCE "seq_z6_06_id_klienta" MINVALUE 1 MAXVALUE 1000 INCREMENT BY 1 CYCLE  
CACHE 50;
```

```
alter table klient
```

```
  modify id_klienta default (seq_z6_06_id_klienta.nextval);
```

```
CREATE SEQUENCE "seq_z6_06_id_rozmiaru" MINVALUE 1 MAXVALUE 1000 INCREMENT BY 1 CYCLE  
CACHE 50;
```

```
alter table rozmiar
```

```
  modify id_rozmiaru default (seq_z6_06_id_rozmiaru.nextval);
```

```
CREATE SEQUENCE "seq_z6_06_id_rodzaju" MINVALUE 1 MAXVALUE 1000 INCREMENT BY 1 CYCLE  
CACHE 50;
```

```
alter table rodzaj
```

```
  modify id_rodzaju default (seq_z6_06_id_rodzaju.nextval);
```



```
CREATE SEQUENCE "seq_z6_06_id_modelu" MINVALUE 1 MAXVALUE 1000 INCREMENT BY 1 CYCLE  
CACHE 50;
```

```
alter table model  
  modify id_modelu default (seq_z6_06_id_modelu.nextval);
```

```
CREATE SEQUENCE "seq_z6_06_id_materialu" MINVALUE 1 MAXVALUE 1000 INCREMENT BY 1 CYCLE  
CACHE 50;
```

```
alter table material  
  modify id_materialu default (seq_z6_06_id_materialu.nextval);
```

```
CREATE SEQUENCE "seq_z6_06_id_kolekcji" MINVALUE 1 MAXVALUE 1000 INCREMENT BY 1 CYCLE  
CACHE 50;
```

```
alter table kolekcja  
  modify id_kolekcji default (seq_z6_06_id_kolekcji.nextval);
```

```
CREATE SEQUENCE "seq_z6_06_id_produkту" MINVALUE 1 MAXVALUE 1000 INCREMENT BY 1 CYCLE  
CACHE 50;
```

```
alter table produkt  
  modify id_produkту default (seq_z6_06_id_produkту.nextval);
```

4. Wprowadzenie danych do tabel

1. Klient

```
//1  
insert into klient  
(nazwa_klienta, miasto, adres)  
values  
(  
  'Abacki',  
  'Warszawa',  
  'ul. Długa 78/12'  
);
```

```
//2  
insert into klient  
(nazwa_klienta, miasto, adres)  
values
```

```
(  
'Babacki',  
'Warszawa',  
'ul. Miła 23/2'  
);
```

```
//3  
insert into klient  
(nazwa_klienta, miasto, adres)  
values  
(  
'Cabacki',  
'Toruń',  
'ul. Radosna 22/2'  
);
```

```
//4  
insert into klient  
(nazwa_klienta, miasto, adres)  
values  
(  
'Dabacki',  
'Pruszków',  
'ul. Jana Olbrachta 13'  
);
```

```
//5  
insert into klient  
(nazwa_klienta, miasto, adres)  
values  
(  
'Kowalski',  
'Toruń',  
'ul. Radosna 22/2'  
);
```

```
//6  
insert into klient  
(nazwa_klienta, miasto, adres)  
values  
(  
'Hotel Zamkowy',  
'Toruń',  
'ul. Podmurna 24'  
);
```

```
//7
insert into klient
(nazwa_klienta, miasto, adres)
values
(
'Hotel Monet',
'Toruń',
'ul. Leśna 24'
);
```

```
//8
insert into klient
(nazwa_klienta, miasto, adres)
values
(
'Leśniczówka',
'Złotoria',
'ul. Daleka 124'
);
```

```
//9
insert into klient
(nazwa_klienta, miasto, adres)
values
(
'Queen',
'Złotoria',
'ul. Poległa 14'
);
```

```
//9
insert into klient
(nazwa_klienta, miasto, adres)
values
(
'Jogolot',
'Turzno',
'ul. Iglasta 14'
);
```

```
//10
insert into klient
(nazwa_klienta, miasto, adres)
values
(
'Queen',
'Złotoria',
'ul. Poległa 14'
);
```

```
//11
insert into klient
(nazwa_klienta, miasto, adres)
values
(
'NOCH',
'Toruń',
'ul. św. Emilii 102/3'
);
```

```
//12
insert into klient
(nazwa_klienta, miasto, adres)
values
(
'DamgMar',
'Toruń',
'ul. Kartonowa 2/12'
);
```

```
//13
insert into klient
(nazwa_klienta, miasto, adres)
values
(
'Mister Ron',
'Kowalewo Pomorskie',
'ul. Witaminowa 20/1'
);
```

```
//14
insert into klient
```

```
(nazwa_klienta, miasto, adres)
values
(
'Domki Miodzio',
'Kamionki',
'ul. Jeziorna 5'
);
```

```
//15
insert into klient
(nazwa_klienta, miasto, adres)
values
(
'Hostel Dworcowy',
'Kowalewo Pomorskie',
'ul. Kolejowa 10/5'
);
```

2. Kolekcja

```
//1
insert into kolekcja
(nazwa_kolekcji, rok, projektant)
values
(
'Classy',
2015,
'Witkowski Michał'
);
```

```
//2
insert into kolekcja
(nazwa_kolekcji, rok, projektant)
values
(
'Modern',
2015,
'Nosál Adam'
);
```

```
//3
insert into kolekcja
(nazwa_kolekcji, rok, projektant)
values
```

```
(  
'Vintage',  
2015,  
'Witkowski Michał'  
);
```

```
//4  
insert into kolekcja  
(nazwa_kolekcji, rok, projektant)  
values  
(  
'Sonata',  
2015,  
'Kostka Robert'  
);
```

```
//5  
insert into kolekcja  
(nazwa_kolekcji, rok, projektant)  
values  
(  
'Contra',  
2015,  
'Nosal Adam'  
);
```

```
//6  
insert into kolekcja  
(nazwa_kolekcji, rok, projektant)  
values  
(  
'Forte',  
2016,  
'Miłto Aneta'  
);
```

3. Rodzaj

```
//1
insert into rodzaj
(nazwa_rodzaju, przeznaczenie)
values
(
'Sofa',
'Salon'
);

//2
insert into rodzaj
(nazwa_rodzaju, przeznaczenie, gwarancja)
values
(
'Narożnik',
'Salon',
30
);

//3
insert into rodzaj
(nazwa_rodzaju, przeznaczenie)
values
(
'Fotel',
'Salon'
);

//4
insert into rodzaj
(nazwa_rodzaju, przeznaczenie, gwarancja)
values
(
'Pufa',
'Salon',
12
);

//5
insert into rodzaj
```

```
(nazwa_rodzaju, przeznaczenie, gwarancja)
values
(
'Podnózek',
'Salon',
12
);
```

```
//6
insert into rodzaj
(nazwa_rodzaju, przeznaczenie)
values
(
'Ława',
'Salon'
);
```

```
//7
insert into rodzaj
(nazwa_rodzaju, przeznaczenie, gwarancja)
values
(
'Fotel biurowy',
'Biuro',
18
);
```

```
//8
insert into rodzaj
(nazwa_rodzaju, przeznaczenie)
values
(
'Łóżko',
'Sypialna'
);
```


4. Rozmiar

*/*Notatki do którego rodzaju dany wymiar się najlepiej nadaje były przydatne do wybierania rozmiaru dla modelu, w rzeczywistym świecie, osoba sprzedająca meble dokładnie wiedziała by jakie rozmiary może przyjmować np. Sofa i wybrała by odpowiednie z listy (lub wprowadziła nowe)*/*

```
//Sofa
//1
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    150,60,60
);
```

```
//Sofa
//2
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    170,60,60
);
```

```
//Sofa
//3
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    250,60,60
);
```

```
//Narożnik, łóżko
//4
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    180,60,150
);
```

```
//Narożnik, łóżko
//5
```

```
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    230,60,200
);
```

```
//Narożnik, łóżko
//6
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    250,50,200
);
```

```
//Fotel
//7
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    100,120,100
);
```

```
//Fotel
//8
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    90,140,100
);
```

```
//Pufa, Podnózek, ława
//9
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    100,60,100
);
```

```
//Pufa, Podnózek, ława
//10
```

```
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    90,60,80
);
```

```
//Pufa, Podnózek, ława
//11
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    70,40,50
);
```

```
//Fotel biurowy
//12
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    60,120,60
);
```

```
//Fotel biurowy
//13
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    70,140,60
);
```

```
//Fotel biurowy
//14
insert into rozmiar
(wymiar_x, wymiar_y, wymiar_z)
values
(
    70,150,60
);
```

5. Model

//Dla każdej kolekcji jest wprowadzane kilka modeli

//Classy

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Ania', 1, 1, 1 //sofa
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Amanda', 1, 1, 3 //sofa
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Ala', 1, 3, 7//Fotel
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Alicja', 1, 4, 10//Pufa
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Alina', 1, 4, 9//Pufa
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Alona', 1, 5, 11//Podnózek
);
```

//Modern

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Paris', 2, 2, 4//Narożnik
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Berlin', 2, 2, 6//Narożnik
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'London', 2, 6, 11//Ława
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'NewYork', 2, 7, 12//Fotel biurowy
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Boston', 2, 7, 14//Fotel biurowy
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Atlanta', 2, 3, 13//Fotel
);
```

//Vintage

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Aster', 3, 1, 2//Sofa
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Jaskier', 3, 1, 1//Sofa
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Róża', 3, 3, 8//Fotel
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Lilia', 3, 8, 4//Łóżko
);
```

//Sonata

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Maria', 4, 2, 5//Narożnik
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Monika', 4, 8, 6//Łóżko
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Magdalena', 4, 6, 9//Ława
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Małorzata', 4, 6, 11//Ława
);
```

//Contra

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Natalia', 5, 1, 2//Sofa
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Nina', 5, 3, 8//Fotel
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Natasza', 5, 7, 13//Fotel biurowy
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Nora', 5, 7, 12//Fotel Biurowy
);
```

// Forte

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
    rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
    'Wanda', 6, 1, 1//Sofa
);
```



```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
      rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
'Weronika', 6, 3, 7//Fotel
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
      rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
'Wiktoria', 6, 5, 10//Podnózek
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
      rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
'Wiola', 6, 5, 11//Podnózek
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
      rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
'Wioletta', 6, 8, 5//Łóżko
);
```

```
insert into model
(nazwa_modelu, kolekcja_id_kolekcji,
      rodzaj_id_rodzaju, rozmiar_id_rozmiaru)
values
(
'Waleria', 6, 7, 12//Fotel biorowy
);
```

6. Materiał

```
//1
insert into material
(nazwa_materialu, kolor)
values
(
'Skóra',
'Czarny'
);
```

```
//2
insert into material
(nazwa_materialu, kolor)
values
(
'Skóra',
'Czerwony'
);
```

```
//3
insert into material
(nazwa_materialu, kolor)
values
(
'Skóra',
'Brązowy'
);
```

```
//4
insert into material
(nazwa_materialu, kolor)
values
(
'Skóra',
'Kremowy'
);
```

```
//5
insert into material
(nazwa_materialu, wzor, kolor)
values
(
'Skóra',
'Karbowany',
'Czarny'
);
```

```
//6
insert into material
(nazwa_materialu,wzor, kolor)
values
(
'Skóra',
'Karbowany',
'Brązowy'
);
```

```
//7
insert into material
(nazwa_materialu, kolor)
values
(
'Zamsz',
'Czarny'
);
```

```
//8
insert into material
(nazwa_materialu, kolor)
values
(
'Zamsz',
'Brązowy'
);
```

```
//9
insert into material
(nazwa_materialu,wzor, kolor)
values
(
'Zamsz',
'Kwiatowy',
```

```
'Brązowy'  
);
```

```
//10  
insert into material  
(nazwa_materialu,wzor, kolor)  
values  
(  
'Zamsz',  
'Geometryczny',  
'Brązowy'  
);
```

```
//11  
insert into material  
(nazwa_materialu,wzor, kolor)  
values  
(  
'Tkanina',  
'Kwiatowy',  
'Czerwony'  
);
```

```
//12  
insert into material  
(nazwa_materialu,wzor, kolor)  
values  
(  
'Tkanina',  
'Kwiatowy',  
'Brązowy'  
);
```

```
//12  
insert into material  
(nazwa_materialu,wzor, kolor)  
values  
(  
'Tkanina',  
'Geometryczny',  
'Czerwony'  
);
```

```
//14  
insert into material
```

```
(nazwa_materialu, kolor)
values
(
'Tkanina',
'Czerwony'
);
```

```
//15
insert into material
(nazwa_materialu, kolor)
values
(
'Tkanina',
'Czarny'
);
```

7. Produkt – wprowadzanie procedurą

/ Aby uzyskać efekt, że każdy model można zakupić w każdym z materiałów stworzyłam procedurę wkonującą pętlę w pętli.*

Dodatkowo procedura ta wywołuje funkcję tworzącą nazwę produktu jako zlepek nazwy modelu oraz atrybutów użytego materiału aby każdy produkt miał łatwy do odszyfrowania „kod” czym tak naprawdę jest.

Procedura ta wykorzystuje również funkcję która losuje cenę. Jest to cena „podstawowa” produktu – przypisana modelowi. Przedział losujący cenę różni się zależnie od rodzaju jakim jest dany model.

Na cenę dodatkowo ma wpływ wybrany materiał, jeśli jest to skóra podnosi cenę produktu o 20% natomiast jeśli zamsz to o 10%. Tak jest przypisywana ostateczna cena dla konkretnego produktu./**

/*na początek musiałam zmienić typ danych dotyczących cen ponieważ początkowy model miał zbyt mały przedział*/

```
alter table produkt
modify cena_produktu numeric(10,2);
```

```
alter table faktura_detale
modify cena_zakupu numeric(10,2);
```

```
alter table faktura_naglowek
modify wartosc_faktury numeric(10,2);
```

//funkcja losująca

```

create or replace function fn_losuj
(v_min integer, v_max integer)
returns numeric(10)
begin
declare v_wylosowana integer;
set v_wylosowana = round(v_min +(v_max- v_min)* rand(),2);

return v_wylosowana;
end;

```

//funkcja losująca cenę podstawową produktu biorąc pod uwagę jego rodzaj

```

create or replace function fn_model_cena
(v_id_modelu integer)
returns numeric(10,2)

begin

declare v_model_rodzaj integer;
declare v_model_cena numeric(10,2);

select rodzaj_id_rodzaju into v_model_rodzaj
from model
where id_modelu = v_id_modelu;

case v_model_rodzaj
when 1 then set v_model_cena=fn_losuj(400,3000)
when 2 then set v_model_cena=fn_losuj(1000,6000)
when 3 then set v_model_cena=fn_losuj(200,2000)
when 4 then set v_model_cena=fn_losuj(50,200)
when 5 then set v_model_cena=fn_losuj(30,200)
when 6 then set v_model_cena=fn_losuj(100,500)
when 7 then set v_model_cena=fn_losuj(200,2000)
when 8 then set v_model_cena=fn_losuj(800,6000)
end case;
return v_model_cena;
end;

```

//funkcja tworząca nazwę produktu według opisanego wyżej wzoru

```

create or replace function fn_produkt_nazwa
(v_id_modelu integer,v_id_materiale integer)
returns varchar(20)

begin

declare v_produkt_nazwa varchar(20);
declare v_model_nazwa varchar(20);
declare v_material_nazwa varchar(20);

```

```

declare v_material_wzor varchar(20);
declare v_material_kolor varchar(20);

select nazwa_modelu into v_model_nazwa
from model
where id_modelu = v_id_modelu;

select substring(nazwa_materialu,1,1) into v_material_nazwa
from material
where id_materialu = v_id_materialu;

select substring(wzor,1,2) into v_material_wzor
from material
where id_materialu = v_id_materialu;

select substring(kolor,1,3) into v_material_kolor
from material
where id_materialu = v_id_materialu;

set v_produkt_nazwa =
v_model_nazwa||'_'||v_material_nazwa||'_'||v_material_wzor||v_material_kolor;

return v_produkt_nazwa;
end;

```

//funkcja tworząca cenę biorąc pod uwagę materiał

```

create or replace function fn_cena_m
(v_id_materialu integer,v_cena numeric(10,2))
returns numeric(10,2)
begin
declare v_material_nazwa varchar(20);
declare v_cena_m numeric(10,2);

select nazwa_materialu into v_material_nazwa
from material
where id_materialu = v_id_materialu;
case v_material_nazwa
when 'Skóra' then set v_cena_m = v_cena+(0.2*v_cena)
when 'Zamsz' then set v_cena_m = v_cena+(0.1*v_cena)
else set v_cena_m=v_cena
end case;

return v_cena_m;
end;

```

//funkcja uzupełniająca tabelę produkt wszystkimi możliwymi wariacjami

```

create or replace procedure pr_utworz_produkt
()
begin

```

```

declare v_max_model integer;
declare v_max_material integer;
declare v_nazwa_produkту varchar(20);

declare v_cena numeric(10,2);
declare v_cena_m numeric(10,2);

declare cnt_model integer = 1;
declare cnt_material integer = 1;

select max(id_modelu) into v_max_model
  from model;
select max(id_materialu) into v_max_material
  from material;

while (cnt_model <= v_max_model) loop

    set v_cena=fn_model_cena(cnt_model);

    set cnt_material=1;

    while (cnt_material <= v_max_material) loop
        set v_nazwa_produkту=fn_produkту_nazwa(cnt_model,cnt_material);
        set v_cena_m=fn_cena_m(cnt_material,v_cena);

        // message v_nazwa_produkту || ' ' || v_cena_m to client;
        insert into produkту
            (nazwa_produkту,cena_produkту,model_id_modelu,material_id_materialu)
            values
            (
                v_nazwa_produkту,
                v_cena_m,
                cnt_model,
                cnt_material
            );

            set cnt_material = cnt_material + 1;
        end loop;

    set cnt_model = cnt_model + 1;
    end loop;

end;

//wywołanie funkcji uzupełniającej tabele produkту
begin
    call pr_utworz_produkту
end;

```


5. Generator faktury

*/*Wywołanie procedury generującej fakturę zostaje połączone do wydarzenia wykonującego się co 2 godziny.*/*

```
ALTER PROCEDURE "z6_06"."pr_generowanie_transakcji"()
begin
  /* to do */
  call pr_utworz_fakture();
end
```

// 1. sekwencja tworząca klucze główne dla faktur

```
CREATE SEQUENCE seq_z6_06_id_faktury" MINVALUE 1 MAXVALUE 1000 INCREMENT BY 1 CYCLE
CACHE 50;
```

```
alter table faktura_naglowek
  modify id_faktury default (seq_z6_06_id_faktury.nextval);
```

// 2. funkcja losująca klienta z przedziału ich id

```
create or replace function fn_losuj_klienta()
  returns integer
begin
  declare v_min integer;
  declare v_max integer;
  declare v_wylosowana integer;
  select min(id_klienta), max(id_klienta) into v_min, v_max
  from klient;
  set v_wylosowana = fn_losuj(v_min, v_max);
  return v_wylosowana;
end;
```

// 3. funkcja losująca produkt z przedziału ich id

```
create or replace function fn_losuj_produkt
  ()
  returns integer
begin
  declare v_min integer;
  declare v_max integer;
  declare v_wylosowana integer;
  select min(id_produktu), max(id_produktu) into v_min, v_max
  from produkt;
  set v_wylosowana = fd_losuj(v_min, v_max);
  return v_wylosowana;
end;
```

// 4. funkcja pobierająca cenę wybranego produktu

```
create or replace function fn_pobierz_cena
(v_id_produktu integer)
returns integer
begin
declare v_cena integer;
select cena_produktu into v_cena
from produkt
where id_produktu = v_id_produktu;
return v_cena;
end;
```

// 5. wyzwalacz na insert

/ update-ujący wartość faktury przy dodaniu nowej pozycji do niej
(przy tworzeniu faktury wartość =0) */*

```
create or replace trigger tr_ins_faktura_detale
before insert on faktura_detale
referencing new as new_rec
for each row
begin
update faktura_naglowek
set wartosc_faktury=wartosc_faktury + new_rec.cena_zakupu
where id_faktury = new_rec.faktura_naglowek_id_faktury
end;
```

// 6. funkcja tworząca pozycję:

//losuje produkt, pobiera cenę, losuje ilość produktu,

// mnoży ją przez cenę, i robi insert włączając działanie wyzwalacza

```
create or replace procedure pr_utworz_pozycje
(v_id_faktura integer ,v_id_pozycja integer)
begin
declare v_id_produktu integer;
declare v_cena_produktu numeric(10,2);
declare v_cena_zakupu numeric(10,2);
declare v_liczba integer;

set v_id_produktu=fn_losuj_produkt();
set v_cena_produktu=fn_pobierz_cena(v_id_produktu);
set v_liczba=fn_losuj(1,50);
set v_cena_zakupu= v_cena_produktu * v_liczba;

insert into faktura_detale
```

```

values
(
v_id_faktura,
v_id_pozycja,
v_liczba,
v_cena_zakupu,
v_id_produktu
);

end;

```

// 7. funkcja tworząca nową fakturę:

*/*Przyjmuje obecną datę, losuje klienta, przypisuje wartości 0 – bęziie to zmieniane wyzwalaczem przy dodawaniu pozycji do faktura_detale, następnie jest losowane ile będzie miała pozycji i dodawane są pozycję odpowiednią procedurą wywoływaną w pętli*/*

```

create or replace procedure pr_utworz_fakture
()
begin
declare v_id_faktury integer;
declare v_data date;
declare v_klient integer;
declare v_wartosc_faktury integer;

declare cnt_pozycja integer;
declare max_pozycja integer;

set v_data = current date;
set v_klient=fn_losuj_klienta();
set v_wartosc_faktury=0;

insert into faktura_naglowek
(data_faktury, wartosc_faktury,klient_id_klienta)
values
(
v_data,
v_wartosc_faktury,
v_klient
);

select max(id_faktury) into v_id_faktury
from faktura_naglowek;

set max_pozycja=fn_losuj(1,10);
set cnt_pozycja =1;

```

```

while (cnt_pozycja <= max_pozycja ) loop
    call pr_utworz_pozycje(v_id_faktury, cnt_pozycja);
    set cnt_pozycja = cnt_pozycja + 1;
end loop;

end;

```

6. Perspektywy klasyczne

1. Perspektywa koniec_gwarancji

/ perspektywa mająca na celu pokazywać dla każdego klienta okres gwarancji zakupionych produktów, klient zazwyczaj bardziej patrzy na rodzaj mebla ewentualnie jego ogólny wygląd (z jakiej kolekcji pochodzi) niż że ten produkt miał jakąś tam nazwę.*

*Perspektywa pokazuje tylko produkty które nie straciły swojej gwarancji. */*

```

create or replace view koniec_gwarancji as
select faktura_naglowek.klient_id_klienta as [klient], klient.nazwa_klienta as [nazwa klienta],
    rodzaj.nazwa_rodzaju as [rodzaj],
    kolekcja.nazwa_kolekcji as [kolekcja], faktura_detale.cena_zakupu as [cena zakupu],
    faktura_naglowek.data_faktury || ' - ' || date(date(data_faktury)+(gwarancja/12)*365) as [okres
gwarancji]
from faktura_naglowek, faktura_detale, produkt, model, kolekcja, rodzaj, klient
where faktura_naglowek.id_faktury=faktura_detale.faktura_naglowek_id_faktury
and faktura_detale.produkt_id_produktu=produkt.id_produktu
and produkt.model_id_modelu=model.id_modelu
and model.rodzaj_id_rodzaju=rodzaj.id_rodzaju
and model.kolekcja_id_kolekcji=kolekcja.id_kolekcji
and faktura_naglowek.klient_id_klienta=klient.id_klienta
and date(date(data_faktury)+(gwarancja/12)*365)>=current date;

```

2. Perspektywa model_o_rozmiarze

/ perspektywa ta ma na celu ułatwić wyszukiwanie mebli po ich wymiarach, stworzona w celu „mam n na m miejsca w salonie, chce tam coś wstawić” dlatego też przedstawia ona modele z przeznaczeniem do salonu z wyłączeniem podnóżka*

Można też dzięki niej wyszukiwać po rodzaju aby dowiedzieć się o wszystkich możliwych modelach (ewentualnie w danym rozmiarze)/*

```

create or replace view model_o_rozmiarze as
select distinct model.nazwa_modelu as [model], rodzaj.nazwa_rodzaju as [rodzaj],
       rozmiar.wymiar_x as [x], rozmiar.wymiar_y as [y], rozmiar.wymiar_z as [z],
       str((rozmiar.wymiar_x*rozmiar.wymiar_z)/10000,3,2) as [powierzchnia m2]
from model, rodzaj, rozmiar
where model.rodzaj_id_rodzaju=rodzaj.id_rodzaju
and model.rozmiar_id_rozmiaru=rozmiar.id_rozmiaru
and nazwa_rodzaju != 'Podnózek'
and przeznaczenie ='Salon';

```

7. Perspektywy zmaterializowane

*/*Odświeżanie perspektyw zostaje podłączone do wydarzenia wykonującego się co 2 godziny które wywołuje procedurę generowania faktur.*/*

```

ALTER PROCEDURE "z6_06"."pr_generowanie_transakcji"()
begin
  /* to do */
  call pr_utworz_fakture();

  REFRESH MATERIALIZED VIEW m_wybrany_material;
  REFRESH MATERIALIZED VIEW m_zarobek_kolekcji;
  REFRESH MATERIALIZED VIEW m_ile_zakupow_klient;

end

```

1. Perspektywa zmaterializowana m_wybrany_material_przed_30

/ perspektywa ta pokaże ile razy został wybrany dany typ materiału (nazwa_materiału) przez klientów, nie biorąc pod uwagę użytych metrów materiału a sam fakt jego „popularności”*

Ta perspektywa zawiera okres czasu do 30.01.2016 z wyłączeniem tego dnia/*

```

create materialized view m_wybrany_material_przed_30 as
select nazwa_materiału as [material], sum(liczba) as [ile razy wybrany]
from faktura_detale, produkt, material, faktura_naglowek
where faktura_detale.produkt_id_produkту= produkt.id_produkту
and produkt.material_id_materialu= material.id_materialu
and faktura_detale.faktura_naglowek_id_faktury=faktura_naglowek.id_faktury
and date(data_faktury) <= date('2016-01-29')

```

group by nazwa_materialu;

REFRESH MATERIALIZED VIEW m_wybrany_material_przed_30;

2. Perspektywa zmaterializowana m_wybrany_material

/ Perspektywa ta działa tak jak perspektywa m_wybrany_material_przed_30 z tym że bierze pod uwagę cały okres działalności sklepu */*

```
create materialized view m_wybrany_material as
select nazwa_materialu as [material] ,sum(liczba) as [ile razy wybrany]
  from faktura_detale, produkt, material,faktura_naglowek
 where faktura_detale.produkt_id_produktu= produkt.id_produktu
 and   produkt.material_id_materialu= material.id_materialu
 and   faktura_detale.faktura_naglowek_id_faktury=faktura_naglowek.id_faktury
 group by nazwa_materialu;
```

REFRESH MATERIALIZED VIEW m_wybrany_material;

3. Perspektywa zmaterializowana m_zarobek_kolekcji

/ W tej perspektywie będzie pokazywane ile dana kolekcja zarobiła od początku działalności sklepu*

Dzięki temu można wyodrębnić np najlepszą i najgorszą kolekcję. I później wykorzystać taką informację przy wyborze projektantów czy tworzeniu nowych kolekcji/*

```
create materialized view m_zarobek_kolekcji as
select nazwa_kolekcji as [kolekcja] ,sum(cena_zakupu) as [zarobiona suma]
  from faktura_detale, produkt, model, kolekcja
 where faktura_detale.produkt_id_produktu=produkt.id_produktu
 and   produkt.model_id_modelu=model.id_modelu
 and   model.kolekcja_id_kolekcji=kolekcja.id_kolekcji
 group by nazwa_kolekcji;
```

REFRESH MATERIALIZED VIEW m_zarobek_kolekcji;

4. Perspektywa zmaterializowana m_ile_zakupów_klient

/ Tutaj zostanie pokazane ile różnych produktów kupił klient . Brane jest jednak pod uwagę uproszczenie że drugi raz kupiony produkt (czy to w tej samej czy innej fakturze) traktowany jest jako inny produkt. Liczenie odbywa się poprzez zliczanie pozycji w fakturach na teo klienta Nie jest tutaj brana pod uwagę ilość zakupionych produktów . */*

```
create materialized view m_ile_zakupow_klient as
select nazwa_klienta as [klient] ,count(pozycja) as [ile zakupów]
  from faktura_detale, faktura_naglowek, klient
 where faktura_detale.faktura_naglowek_id_faktury=faktura_naglowek.id_faktury
 and faktura_naglowek.klient_id_klienta=klient.id_klienta
 group by nazwa_klienta;
```

```
REFRESH MATERIALIZED VIEW m_ile_zakupow_klient;
```

4. Skrypty wdrożeniowe instalujące i deinstalujące projekt

Skrypt wdrożeniowy został pokazany w krokach w punkcie 3 dokumentacji.

Skrypt deinstalujący:

```
drop function fn_losuj;
drop function fn_model_cena;
drop function fn_produkt_nazwa;
drop function fn_cena_m;
drop procedure pr_utworz_produkt;
```

```
drop function fn_losuj_klienta;
drop function fn_losuj_produkt;
drop function fn_pobierz_cene;
```

```
drop trigger tr_ins_faktura_detale;
```

```
drop procedure pr_utworz_pozycje;
drop procedure pr_utworz_fakture;
```

```
drop sequence seq_z6_06_id_klienta;  
drop sequence seq_z6_06_id_rozmiaru;  
drop sequence seq_z6_06_id_rodzaju;  
drop sequence seq_z6_06_id_modelu;  
drop sequence seq_z6_06_id_materialu;  
drop sequence seq_z6_06_id_produktu;  
drop sequence seq_z6_06_id_faktury;
```

```
drop view koniec_gwarancji;  
drop view model_o_rozmiarze;
```

```
drop materialized view m_ile_zakupow_klient;  
drop materialized view m_zarobek_kolekcji;  
drop materialized view m_wybrany_material;  
drop materialized view m_wybrany_material_przed_30;
```

```
drop table faktura_naglowek;  
drop table faktura_detale;  
drop table klient;  
drop table produkt;  
drop table material;  
drop table model;  
drop table kolekcja;  
drop table rodzaj;  
drop table rozmiar;
```

5. Instrukcja instalacji projektu i sprawdzenia jej poprawności

1. Instrukcja instalacji

Aby zainstalować projekt należy w programie Synbase Central zalogować się na serwer posiadanymi danymi. Następnie w programie Interactive SQL należy kolejno wklejać fragmenty skryptu podane w odpowiednich podpunktach punktu 3 dokumentacji oraz uruchamiać ich wykonanie. Należy zwrócić uwagę aby nie przekleić tytułu podpunktu. Po wykonaniu tego baza powinna być poprawnie zainstalowana na serwerze.

Jeśli jednak istnieje możliwość wygodniejszym rozwiązaniem było by skorzystanie w wcześniej przygotowanych ponumerowanych plików txt zawierających odpowiednie fragmenty skryptu aby wyeliminować ryzyko przeklejenia tekstu nie będącego kodem czy komentarzem lub ucięcia części kodu.

Nowe faktury będą tworzyć się co 2 godziny na koncie z6_06 gdyby zaszła potrzeba stworzenia nowych faktur należy wywołać funkcję call pr_utworz_fakture();

2. Sprawdzenie poprawności

Sprawdzenie poprawności zainstalowania projektu dokonuje się poprzez wywołanie zawartości tabel odpowiednimi instrukcjami. Np:

//wyswietlenie zawartości kilku tablic dla sprawdzenia poprawności dodawania danych do tabel

//wyswietlenie tablicy produkt – czy procedura tworząca działa poprawnie, sprawdza też poprawne dodanie materiałów i modeli

```
select * from produkt
order by id_produktu;
```

//wyswietlenie tablicy faktura_detale – czy generator faktur działa poprawnie

```
select * from faktura_detale
order by faktura_naglowek_id_faktury, pozycja;
```

//sprawdzenie zawartości perspektyw

//koniec_gwarancji sięga po dane z tabel faktura_naglowek, faktura_detale, produkt, model, kolekcja,

//rodzaj, klient dzięki czemu możemy zobaczyć że większość bazy danych napewno działa poprawnie.

```
select * from koniec_gwarancji
order by klient, [okres gwarancji], kolekcja;
```

```
select * from koniec_gwarancji
where klient =1
order by [okres gwarancji], kolekcja;
```

```
select * from model_o_rozmiarze;
select * from model_o_rozmiarze
where (([x] between 150 and 200 and [z] <=80) or [powierzchnia m2]<1)
and upper([rodzaj])=upper('Ława');
```

//wyswietlenie danych perspektyw zmaterializowanych

```
select * from m_wybrany_material_przed_30;
select * from m_wybrany_material;
```

```
select * from m_zarobek_kolekcji;
select round(avg([zarobiona suma]),2) as [srednio zarobienie przez kolekcje]
from m_zarobek_kolekcji;
```

```
select * from m_ile_zakupow_klient
order by [ile zakupów] desc;
```

6. Wyniki działania perspektyw analitycznych w postaci zrzutów ekranowych

asa12_city (z6_06) on asa12_city - Interactive SQL

File Edit SQL Data Favorites Tools Window Help

SQL Statements

```
1 select * from koniec_gwarancji
2 order by klient, [okres gwarancji], kolekcja;
3
```

Results

	klient	nazwa klienta	rodzaj	kolekcja	cena zakupu	okres gwarancji
1	1	Abacki	Pufa	Classy	5 792,00	2016-01-29 - 2017-01-28
2	1	Abacki	Fotel biurowy	Contra	65 870,00	2016-01-29 - 2017-07-29
3	1	Abacki	Fotel biurowy	Modern	35 105,00	2016-01-29 - 2017-07-29
4	1	Abacki	Sofa	Classy	21 504,00	2016-01-29 - 2018-01-28
5	1	Abacki	Sofa	Forte	78 554,00	2016-01-29 - 2018-01-28
6	1	Abacki	Fotel	Vintage	26 688,00	2016-01-29 - 2018-01-28
7	2	Babacki	Sofa	Classy	1 768,00	2016-01-30 - 2018-01-29
8	3	Cabacki	Podnózek	Forte	1 023,00	2016-01-29 - 2017-01-28
9	3	Cabacki	Sofa	Classy	27 404,00	2016-01-29 - 2018-01-28
10	3	Cabacki	Sofa	Classy	13 824,00	2016-01-29 - 2018-01-28
11	3	Cabacki	Sofa	Classy	14 246,00	2016-01-29 - 2018-01-28
12	3	Cabacki	Fotel	Classy	785,00	2016-01-29 - 2018-01-28
13	3	Cabacki	Sofa	Contra	5 274,00	2016-01-29 - 2018-01-28
14	3	Cabacki	Fotel	Vintage	23 352,00	2016-01-29 - 2018-01-28
15	3	Cabacki	Fotel	Vintage	25 480,00	2016-01-29 - 2018-01-28
16	3	Cabacki	Łóżko	Vintage	157 260,00	2016-01-29 - 2018-01-28
17	4	Dabacki	Sofa	Contra	59 950,00	2016-01-29 - 2018-01-28
18	4	Dabacki	Łóżko	Forte	166 200,00	2016-01-29 - 2018-01-28
19	4	Dabacki	Narożnik	Sonata	50 008,00	2016-01-29 - 2018-07-29
20	6	Hotel Zamkowy	Fotel biurowy	Modern	35 105,00	2016-01-29 - 2017-07-29
21	6	Hotel Zamkowy	Fotel biurowy	Modern	37 620,00	2016-01-29 - 2017-07-29
22	6	Hotel Zamkowy	Sofa	Contra	86 310,00	2016-01-29 - 2018-01-28
23	6	Hotel Zamkowy	Fotel	Modern	16 928,00	2016-01-29 - 2018-01-28
24	6	Hotel Zamkowy	Fotel	Modern	50 784,00	2016-01-29 - 2018-01-28
25	6	Hotel Zamkowy	Ława	Modern	4 266,00	2016-01-29 - 2018-01-28
26	6	Hotel Zamkowy	Ława	Modern	3 476,00	2016-01-29 - 2018-01-28
27	6	Hotel Zamkowy	Sofa	Vintage	21 350,00	2016-01-29 - 2018-01-28
28	7	Hotel Monet	Podnózek	Forte	6 478,00	2016-01-29 - 2017-01-28
29	7	Hotel Monet	Fotel biurowy	Contra	56 450,00	2016-01-29 - 2017-07-29
30	7	Hotel Monet	Łóżko	Forte	174 510,00	2016-01-29 - 2018-01-28
31	7	Hotel Monet	Fotel	Modern	11 638,00	2016-01-29 - 2018-01-28
32	7	Hotel Monet	Fotel	Modern	35 972,00	2016-01-29 - 2018-01-28
33	7	Hotel Monet	Ława	Sonata	4 692,00	2016-01-29 - 2018-01-28
34	9	Queen	Pufa	Classy	2 940,00	2016-01-29 - 2017-01-28
35	9	Queen	Pufa	Classy	6 154,00	2016-01-29 - 2017-01-28
36	9	Queen	Pufa	Classy	995,00	2016-01-29 - 2017-01-28
37	9	Queen	Fotel biurowy	Forte	60 225,00	2016-01-29 - 2017-07-29
38	9	Queen	Fotel	Contra	6 740,00	2016-01-29 - 2018-01-28

Results Messages

Line 3 Column 1 64 rows

asa12_city (z6_06) on asa12_city - Interactive SQL

File Edit SQL Data Favorites Tools Window Help

SQL Statements

```

1 select * from koniec_gwarancji
2 where klient =1
3 order by [okres gwarancji], kolekcja;
4
5
6
7

```

Results

	klient	nazwa klienta	rodzaj	kolekcja	cena zakupu	okres gwarancji
1	1	Abacki	Pufa	Classy	5 792,00	2016-01-29 - 2017-01-28
2	1	Abacki	Fotel biurowy	Contra	65 870,00	2016-01-29 - 2017-07-29
3	1	Abacki	Fotel biurowy	Modern	35 105,00	2016-01-29 - 2017-07-29
4	1	Abacki	Sofa	Classy	21 504,00	2016-01-29 - 2018-01-28
5	1	Abacki	Sofa	Forte	78 554,00	2016-01-29 - 2018-01-28
6	1	Abacki	Fotel	Vintage	26 688,00	2016-01-29 - 2018-01-28

asa12_city (z6_06) on asa12_city - Interactive SQL

File Edit SQL Data Favorites Tools Window Help

SQL Statements

```
1 select * from model_o_rozmiarze;
```

Results

	model	rodzaj	x	y	z	powierzchnia m2
1	Ala	Fotel	100	120	100	1.0
2	Alicja	Pufa	90	60	80	0.7
3	Alina	Pufa	100	60	100	1.0
4	Amanda	Sofa	250	60	60	1.5
5	Ania	Sofa	150	60	60	0.9
6	Aster	Sofa	170	60	60	1.0
7	Atlanta	Fotel	70	140	60	0.4
8	Berlin	Narożnik	250	50	200	5.0
9	Jaskier	Sofa	150	60	60	0.9
10	London	Ława	70	40	50	0.3
11	Magdalena	Ława	100	60	100	1.0
12	Maria	Narożnik	230	60	200	4.6
13	Malorzata	Ława	70	40	50	0.3
14	Natalia	Sofa	170	60	60	1.0
15	Nina	Fotel	90	140	100	0.9
16	Paris	Narożnik	180	60	150	2.7
17	Róża	Fotel	90	140	100	0.9
18	Wanda	Sofa	150	60	60	0.9
19	Weronika	Fotel	100	120	100	1.0

Results Messages

Line 2 Column 1 19 rows

asa12_city (z6_06) on asa12_city - Interactive SQL

File Edit SQL Data Favorites Tools Window Help

SQL Statements

```

1 select * from model_o_rozmiarze
2   where (([x] between 150 and 200 and [z] <=80) or [powierzchnia m2]<1)
3   and upper([rodzaj])=upper('Ława');
4
5
6
7

```

Results

	model	rodzaj	x	y	z	powierzchnia m2
1	London	Ława	70	40	50	0.3
2	Małorzata	Ława	70	40	50	0.3

asa12_city (z6_06) on asa12_city - Interactive SQL

File Edit SQL Data Favorites Tools Window Help

SQL Statements

```

1 select * from m_wybrany_material_przed_30;
2

```

Results

	material	ile razy wybrany
1	Skóra	503
2	Zamsz	224
3	Tkanina	573

asa12_city (z6_06) on asa12_city - Interactive SQL

File Edit SQL Data Favorites Tools Window Help

SQL Statements

```

1 select * from m_wybrany_material;
2

```

Results

	material	ile razy wybrany
1	Skóra	595
2	Zamsz	399
3	Tkanina	710

asa12_city (z6_06) on asa12_city - Interactive SQL

File Edit SQL Data Favorites Tools Window Help

SQL Statements

```
1
2 select * from m_zarobek_kolekcji;
```

Results

	kolekcja	zarobiona suma
1	Forte	943 933,00
2	Contra	481 691,00
3	Classy	184 561,00
4	Modern	323 888,00
5	Vintage	565 163,00
6	Sonata	265 204,00

asa12_city (z6_06) on asa12_city - Interactive SQL

File Edit SQL Data Favorites Tools Window Help

SQL Statements

```
1 select round(avg([zarobiona suma]),2) as [srednio zarobienie przez kolekcje]
2 from m_zarobek_kolekcji;
```

Results

	srednio zarobienie przez kolekcje
1	460 740,000000

asa12_city (z6_06) on asa12_city - Interactive SQL

File Edit SQL Data Favorites Tools Window Help

SQL Statements

```
1 select * from m_ile_zakupow_klient
2 order by [ile zakupów] desc;
```

Results

	klient	ile zakupów
1	Domki Miodzio	17
2	Queen	9
3	Cabacki	9
4	Hotel Zamkowy	8
5	Abacki	6
6	Hotel Monet	6
7	Jogolot	5
8	Dabacki	3
9	Babacki	1