Bazy Danych
Sprawozdanie z zadania projektowego

Temat:
Sklep z butami

Autor: ***Hubert Filipczuk***

Polecenie

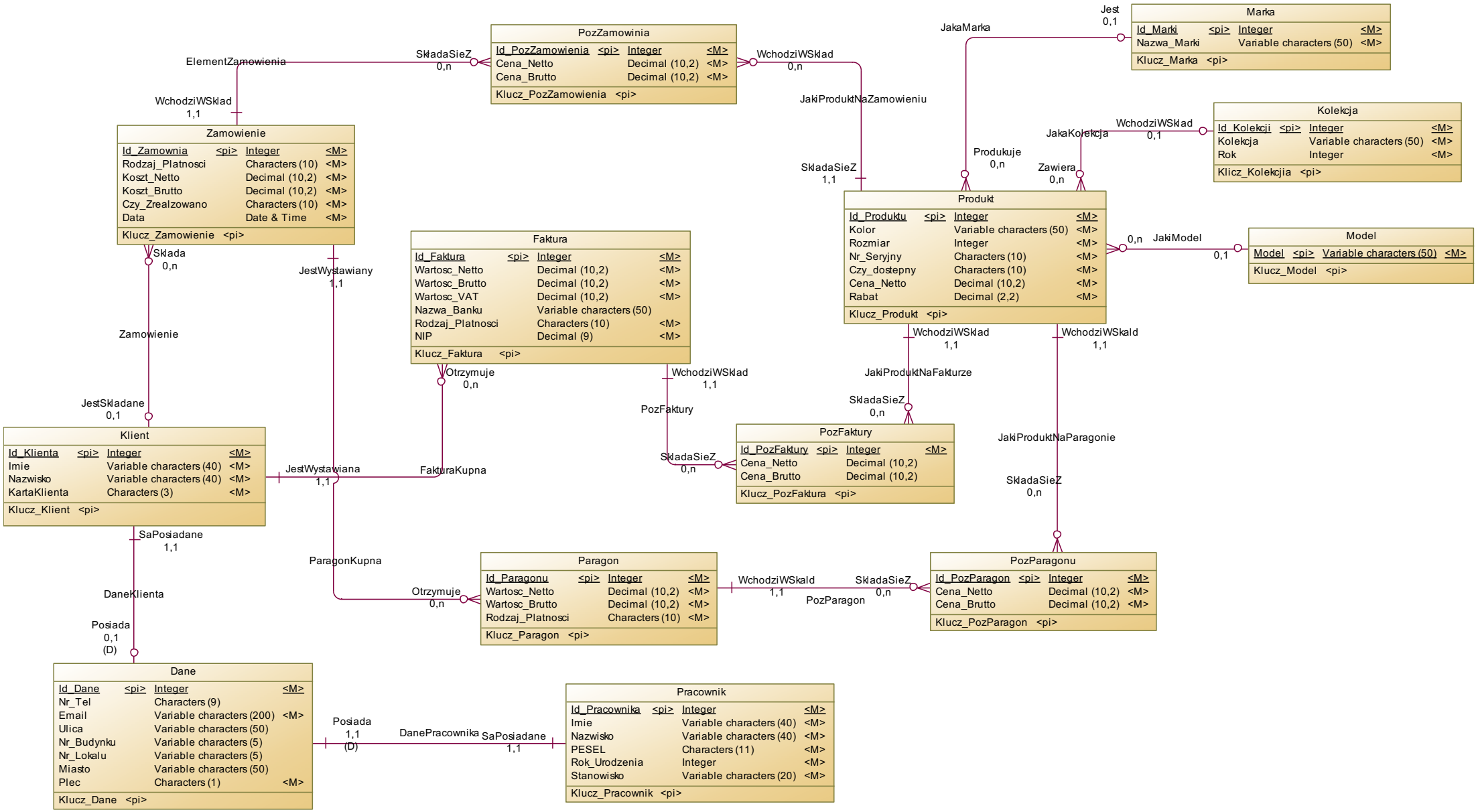
Zaprojektować model oraz stworzyć dwie implantację bazy danych.

Projekt bazy danych prezentuje sklep z butami. Sklep sprzedaje buty znanych producentów, między innymi Nike, Adidas czy Puma. Baza ta pozwala na przechowywanie danych o produktach (butach), czyli jakiej marki jest dany produkt, z jakiej kolekcji i model produktu, ponadto w bazie zawarte są informacje o klientach, zamówieniach oraz baza umożliwia tworzenie i magazynowanie faktur oraz paragonów. Sklep ten jest stworzony dla miłośników butów typu Sneaker. Model bazy danych został stworzony w oparciu o PowerDesigner a do stworzenia samej bazy danych Microsoft SQL Server i MySQL.

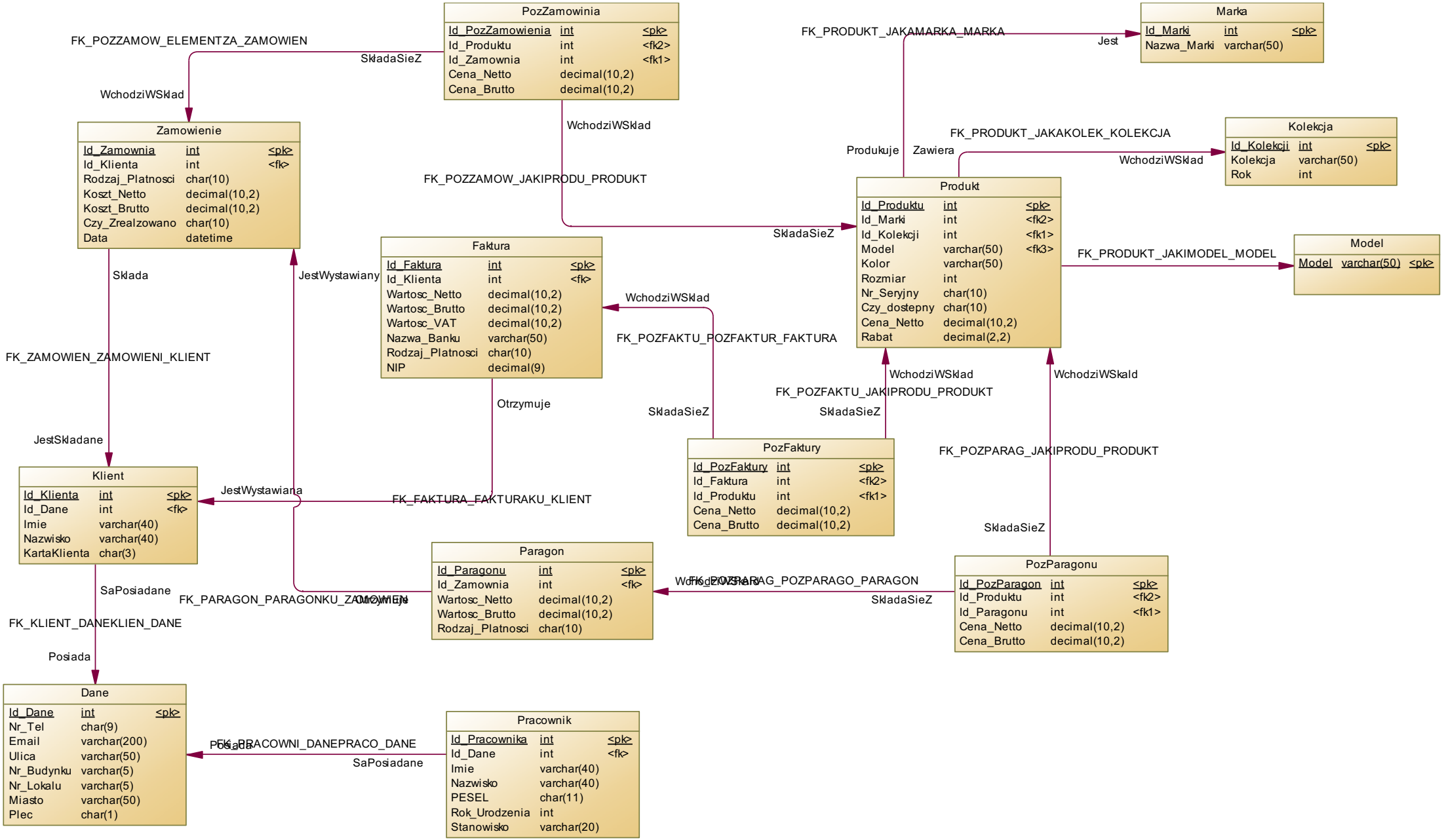
Założenia projektowe

- minimum 10 tabel
- 3 widoki
- 3 procedury
- 3 funkcje
- 3 triggery
- 3 użytkownicy

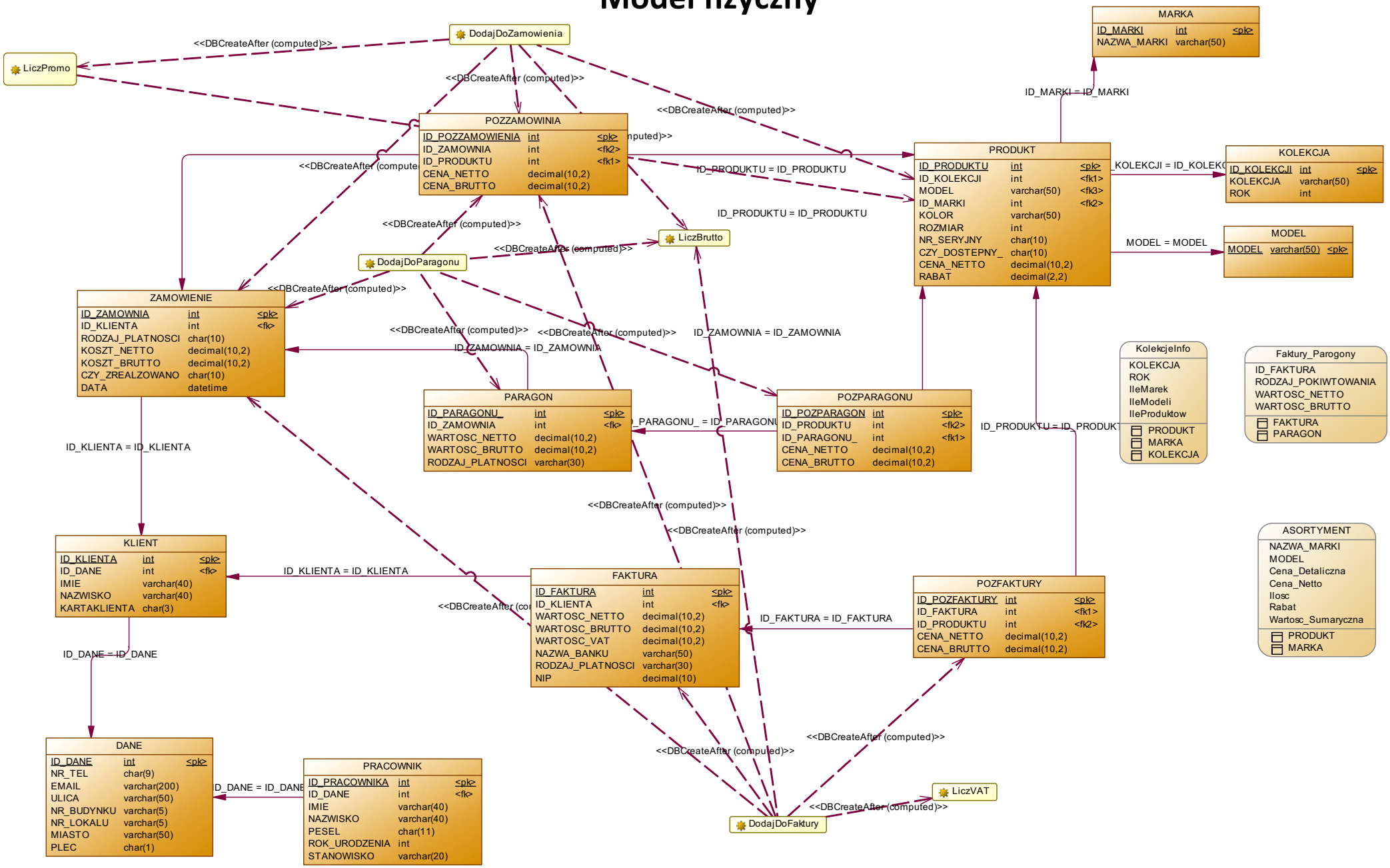
Model Konceptualny



Model fizyczny uproszczony



Model fizyczny



Listing Kodu Bazy

```
/*==============================================================*/
/* Table: DANE                                                  */
/*==============================================================*/
create table DANE (
   ID_DANE              int                  not null,
   NR_TEL               char(9)              null,
   EMAIL                varchar(200)         not null,
   ULICA                varchar(50)          null,
   NR_BUDYNKU           varchar(5)           null,
   NR_LOKALU            varchar(5)           null,
   MIASTO               varchar(50)          null,
   PLEC                 char(1)              not null default 'K'
      constraint CKC_PLEC_DANE check (PLEC in ('K','M','I') and PLEC = upper(PLEC)),
   constraint PK_DANE primary key nonclustered (ID_DANE)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('DANE') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'DANE'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Dodatkowe informacje pozwalaj¹ce na rozpoznanie klientow oraz pracownikow.',
   'user', @CurrentUser, 'table', 'DANE'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('DANE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_DANE')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'ID_DANE'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer danych.',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'ID_DANE'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('DANE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'NR_TEL')
)
begin
```

```
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'NR_TEL'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Ci¹g cyfr identyfikuj¹cych abonenta telefonicznego.',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'NR_TEL'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('DANE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'EMAIL')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'EMAIL'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Us³uga internetowa, w nomenklaturze prawnej okreœlana jako œwiadczenie us³ug
drog¹ elektroniczn¹, s³u¿¹ca do przesy³ania wiadomoœci tekstowych lub
multimedialnych.',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'EMAIL'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('DANE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ULICA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'ULICA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Droga na terenie zabudowanym lub przeznaczonym do zabudowy, g³ównie w mieœcie,
ale tak¿e w osadzie oraz czasem na wsi, której wspó³czeœnie zazwyczaj nadaje siê
oficjalnie urzêdow¹ nazwê w³asn¹.',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'ULICA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('DANE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'NR_BUDYNKU')
```

```
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'NR_BUDYNKU'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Numer danego bloku lub domu na ulicy.',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'NR_BUDYNKU'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('DANE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'NR_LOKALU')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'NR_LOKALU'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Pomieszczenie, w którym masz firmê lub mieszkasz.',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'NR_LOKALU'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('DANE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'MIASTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'MIASTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Historycznie ukszta³towana jednostka osadnicza charakteryzuj¹ca siê du¿¹
intensywnoœci¹ zabudowy, ma³¹ iloœci¹ terenów rolniczych, ludnoœci¹ pracuj¹c¹ poza
rolnictwem (w przemyœle lub w us³ugach) prowadz¹c¹ miejski styl ¿ycia.',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'MIASTO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('DANE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'PLEC')
```

```
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'PLEC'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Zespó³ cech budowy o charakterze struktur, funkcji i zachowañ pozwalaj¹cych na
sklasyfikowanie organizmów na mêskie i ¿eñskie.',
   'user', @CurrentUser, 'table', 'DANE', 'column', 'PLEC'
go

/*==============================================================*/
/* Table: FAKTURA                                               */
/*==============================================================*/
create table FAKTURA (
   ID_FAKTURA           int                  not null,
   ID_KLIENTA           int                  not null,
   WARTOSC_NETTO        decimal(10,2)        not null,
   WARTOSC_BRUTTO       decimal(10,2)        not null,
   WARTOSC_VAT          decimal(10,2)        not null,
   NAZWA_BANKU          varchar(50)          null,
   RODZAJ_PLATNOSCI     char(10)             not null
      constraint CKC_RODZAJ_PLATNOSCI_FAKTURA check (RODZAJ_PLATNOSCI in
('Karta','Gotowka')),
   NIP                  decimal(9)           not null,
   constraint PK_FAKTURA primary key nonclustered (ID_FAKTURA)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('FAKTURA') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'FAKTURA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Faktura jest wystawiana dla klientów.',
   'user', @CurrentUser, 'table', 'FAKTURA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('FAKTURA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_FAKTURA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'ID_FAKTURA'
```

```
end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer danej faktury.',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'ID_FAKTURA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('FAKTURA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_KLIENTA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'ID_KLIENTA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer klienta sklepu.',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'ID_KLIENTA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('FAKTURA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'WARTOSC_NETTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'WARTOSC_NETTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Wartoœæ danego towaru lub us³ugi przed doliczeniem do niej podatku.',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'WARTOSC_NETTO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('FAKTURA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'WARTOSC_BRUTTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'WARTOSC_BRUTTO'

end
```

```
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Wartoœæ towaru lub us³ugi z doliczon¹ kwot¹ podatku VAT.',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'WARTOSC_BRUTTO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('FAKTURA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'WARTOSC_VAT')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'WARTOSC_VAT'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Podatek VAT nale¿ny to kwota zobowi¹zania, która powstaje w momencie wystawienia
faktury sprzeda¿y.',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'WARTOSC_VAT'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('FAKTURA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'NAZWA_BANKU')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'NAZWA_BANKU'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Jest to informacja o tym jak nazywa siê bank przez który zosta³o zrealizowane
zamówienie.',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'NAZWA_BANKU'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('FAKTURA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'RODZAJ_PLATNOSCI')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'RODZAJ_PLATNOSCI'

end
```

```
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Informacja o tym czym zosta³ dokonany zakup: Gotówk¹ lu Kart¹.',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'RODZAJ_PLATNOSCI'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('FAKTURA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'NIP')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'NIP'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Numer Identyfikacji Podatkowej jest dziewiêciocyfrowym kodem, przyporz¹dkowanym
do konkretnego podatnika, nadawanym z urzêdu lub na wniosek.',
   'user', @CurrentUser, 'table', 'FAKTURA', 'column', 'NIP'
go

/*==============================================================*/
/* Index: FAKTURAKUPNA_FK                                       */
/*==============================================================*/
create index FAKTURAKUPNA_FK on FAKTURA (
ID_KLIENTA ASC
)
go

/*==============================================================*/
/* Table: KLIENT                                                */
/*==============================================================*/
create table KLIENT (
   ID_KLIENTA           int                  not null,
   ID_DANE              int                  null,
   IMIE                 varchar(40)          not null,
   NAZWISKO             varchar(40)          not null,
   KARTAKLIENTA         char(3)              not null default 'TAK'
      constraint CKC_KARTAKLIENTA_KLIENT check (KARTAKLIENTA in ('TAK','NIE') and
KARTAKLIENTA = upper(KARTAKLIENTA)),
   constraint PK_KLIENT primary key nonclustered (ID_KLIENTA)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('KLIENT') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'KLIENT'

end
```

```
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Klient jest to osoba sk³adaj¹ca zamówienie w sklepie posiadaj¹ca kartê klienta
lub dokonuj¹ca zakupu z faktur¹.',
   'user', @CurrentUser, 'table', 'KLIENT'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('KLIENT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_KLIENTA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'KLIENT', 'column', 'ID_KLIENTA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer klienta sklepu.',
   'user', @CurrentUser, 'table', 'KLIENT', 'column', 'ID_KLIENTA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('KLIENT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_DANE')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'KLIENT', 'column', 'ID_DANE'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer danych.',
   'user', @CurrentUser, 'table', 'KLIENT', 'column', 'ID_DANE'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('KLIENT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'IMIE')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'KLIENT', 'column', 'IMIE'

end
```

```
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Osobista nazwa nadawana osobie przez grupê, do której nale¿y.',
   'user', @CurrentUser, 'table', 'KLIENT', 'column', 'IMIE'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('KLIENT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'NAZWISKO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'KLIENT', 'column', 'NAZWISKO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Nazwa rodziny, do której dana osoba nale¿y',
   'user', @CurrentUser, 'table', 'KLIENT', 'column', 'NAZWISKO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('KLIENT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'KARTAKLIENTA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'KLIENT', 'column', 'KARTAKLIENTA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Wszystkich istotnych informacji o kontrahencie w jednym miejscu. Pomieœci nie
tylko dane kontaktowe i zdjêcie, ale tak¿e historiê finansow¹ oraz wszelkie inne
treœci - mo¿na j¹ dowolnie dostosowywaæ do w³asnych potrzeb.',
   'user', @CurrentUser, 'table', 'KLIENT', 'column', 'KARTAKLIENTA'
go

/*==============================================================*/
/* Index: DANEKLIENTA_FK                                        */
/*==============================================================*/
create index DANEKLIENTA_FK on KLIENT (
ID_DANE ASC
)
go

/*==============================================================*/
/* Table: KOLEKCJA                                              */
/*==============================================================*/
create table KOLEKCJA (
   ID_KOLEKCJI          int                  not null,
   KOLEKCJA             varchar(50)          not null
```

```
      constraint CKC_KOLEKCJA_KOLEKCJA check (KOLEKCJA in
('Zima','Wiosna','Lato','Jesien')),
   ROK                  int                  not null,
   constraint PK_KOLEKCJA primary key nonclustered (ID_KOLEKCJI)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('KOLEKCJA') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'KOLEKCJA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Jest to informacja o tym, z jakiej kolekcji jest dany produkt:
(Zima/Wiosna/Lato/Jesieñ) + Rok.',
   'user', @CurrentUser, 'table', 'KOLEKCJA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('KOLEKCJA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_KOLEKCJI')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'KOLEKCJA', 'column', 'ID_KOLEKCJI'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer kolekcji.',
   'user', @CurrentUser, 'table', 'KOLEKCJA', 'column', 'ID_KOLEKCJI'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('KOLEKCJA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'KOLEKCJA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'KOLEKCJA', 'column', 'KOLEKCJA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Sezon z jakiej kolekcji pochodzi dany produkt: Zima, Wiosna, Lato, Jesieñ',
```

```
   'user', @CurrentUser, 'table', 'KOLEKCJA', 'column', 'KOLEKCJA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('KOLEKCJA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ROK')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'KOLEKCJA', 'column', 'ROK'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Z którego roku pochodzi dany produkt.',
   'user', @CurrentUser, 'table', 'KOLEKCJA', 'column', 'ROK'
go

/*==============================================================*/
/* Table: MARKA                                                 */
/*==============================================================*/
create table MARKA (
   ID_MARKI             int                  not null,
   NAZWA_MARKI          varchar(50)          not null,
   constraint PK_MARKA primary key nonclustered (ID_MARKI)
)
go

if exists (select 1 from  sys.extended_properties
          where major_id = object_id('MARKA') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'MARKA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Jest to informacjao tym jakiej Marki jest dany produkt.',
   'user', @CurrentUser, 'table', 'MARKA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('MARKA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_MARKI')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'MARKA', 'column', 'ID_MARKI'

end
```

```
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer marki.',
   'user', @CurrentUser, 'table', 'MARKA', 'column', 'ID_MARKI'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('MARKA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'NAZWA_MARKI')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'MARKA', 'column', 'NAZWA_MARKI'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Nazwa jak¹ ma dana marka produktu.',
   'user', @CurrentUser, 'table', 'MARKA', 'column', 'NAZWA_MARKI'
go

/*==============================================================*/
/* Table: MODEL                                                 */
/*==============================================================*/
create table MODEL (
   MODEL                varchar(50)          not null,
   constraint PK_MODEL primary key nonclustered (MODEL)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('MODEL') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'MODEL'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Jest to nazwa modelu danych butów.',
   'user', @CurrentUser, 'table', 'MODEL'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('MODEL')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'MODEL')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
```

```
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'MODEL', 'column', 'MODEL'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Nazwa modelu z jakiej pochodzi dany produkt.',
   'user', @CurrentUser, 'table', 'MODEL', 'column', 'MODEL'
go

/*==============================================================*/
/* Table: PARAGON                                               */
/*==============================================================*/
create table PARAGON (
   ID_PARAGONU_         int                  not null,
   ID_ZAMOWNIA          int                  not null,
   WARTOSC_NETTO        decimal(10,2)        not null,
   WARTOSC_BRUTTO       decimal(10,2)        not null,
   RODZAJ_PLATNOSCI     char(10)             not null
      constraint CKC_RODZAJ_PLATNOSCI_PARAGON check (RODZAJ_PLATNOSCI in
('Karta','Gotowka')),
   constraint PK_PARAGON primary key nonclustered (ID_PARAGONU_)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('PARAGON') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PARAGON'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Dokument potwierdzaj¹cy dokonanie zakupu.',
   'user', @CurrentUser, 'table', 'PARAGON'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PARAGON')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_PARAGONU_')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PARAGON', 'column', 'ID_PARAGONU_'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer paragonu',
   'user', @CurrentUser, 'table', 'PARAGON', 'column', 'ID_PARAGONU_'
```

```
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PARAGON')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_ZAMOWNIA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PARAGON', 'column', 'ID_ZAMOWNIA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer zamówienia.',
   'user', @CurrentUser, 'table', 'PARAGON', 'column', 'ID_ZAMOWNIA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PARAGON')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'WARTOSC_NETTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PARAGON', 'column', 'WARTOSC_NETTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Wartoœæ danego towaru lub us³ugi przed doliczeniem do niej podatku.',
   'user', @CurrentUser, 'table', 'PARAGON', 'column', 'WARTOSC_NETTO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PARAGON')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'WARTOSC_BRUTTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PARAGON', 'column', 'WARTOSC_BRUTTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Wartoœæ towaru lub us³ugi z doliczon¹ kwot¹ podatku VAT.',
   'user', @CurrentUser, 'table', 'PARAGON', 'column', 'WARTOSC_BRUTTO'
go
```

```
if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PARAGON')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'RODZAJ_PLATNOSCI')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PARAGON', 'column', 'RODZAJ_PLATNOSCI'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Informacja o tym czym zosta³ dokonany zakup: Gotówk¹ lu Kart¹.',
   'user', @CurrentUser, 'table', 'PARAGON', 'column', 'RODZAJ_PLATNOSCI'
go

/*==============================================================*/
/* Index: PARAGONKUPNA_FK                                       */
/*==============================================================*/
create index PARAGONKUPNA_FK on PARAGON (
ID_ZAMOWNIA ASC
)
go

/*==============================================================*/
/* Table: POZFAKTURY                                            */
/*==============================================================*/
create table POZFAKTURY (
   ID_POZFAKTURY        int                  not null,
   ID_FAKTURA           int                  not null,
   ID_PRODUKTU          int                  not null,
   CENA_NETTO           decimal(10,2)        null,
   CENA_BRUTTO          decimal(10,2)        null,
   constraint PK_POZFAKTURY primary key nonclustered (ID_POZFAKTURY)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('POZFAKTURY') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZFAKTURY'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Pozycja Faktury',
   'user', @CurrentUser, 'table', 'POZFAKTURY'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZFAKTURY')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_POZFAKTURY')
```

```
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZFAKTURY', 'column', 'ID_POZFAKTURY'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer pozycji faktury.',
   'user', @CurrentUser, 'table', 'POZFAKTURY', 'column', 'ID_POZFAKTURY'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZFAKTURY')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_FAKTURA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZFAKTURY', 'column', 'ID_FAKTURA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer danej faktury.',
   'user', @CurrentUser, 'table', 'POZFAKTURY', 'column', 'ID_FAKTURA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZFAKTURY')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_PRODUKTU')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZFAKTURY', 'column', 'ID_PRODUKTU'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer produktu.',
   'user', @CurrentUser, 'table', 'POZFAKTURY', 'column', 'ID_PRODUKTU'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZFAKTURY')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'CENA_NETTO')
)
begin
```

```
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZFAKTURY', 'column', 'CENA_NETTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Cena danego towaru lub us³ugi przed doliczeniem do niej podatku.',
   'user', @CurrentUser, 'table', 'POZFAKTURY', 'column', 'CENA_NETTO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZFAKTURY')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'CENA_BRUTTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZFAKTURY', 'column', 'CENA_BRUTTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Cena towaru lub us³ugi z doliczon¹ kwot¹ podatku VAT.',
   'user', @CurrentUser, 'table', 'POZFAKTURY', 'column', 'CENA_BRUTTO'
go

/*==============================================================*/
/* Index: JAKIPRODUKTNAFAKTURZE_FK                              */
/*==============================================================*/
create index JAKIPRODUKTNAFAKTURZE_FK on POZFAKTURY (
ID_PRODUKTU ASC
)
go

/*==============================================================*/
/* Index: POZFAKTURY_FK                                         */
/*==============================================================*/
create index POZFAKTURY_FK on POZFAKTURY (
ID_FAKTURA ASC
)
go

/*==============================================================*/
/* Table: POZPARAGONU                                           */
/*==============================================================*/
create table POZPARAGONU (
   ID_POZPARAGON        int                  not null,
   ID_PRODUKTU          int                  not null,
   ID_PARAGONU_         int                  not null,
   CENA_NETTO           decimal(10,2)        not null,
   CENA_BRUTTO          decimal(10,2)        not null,
   constraint PK_POZPARAGONU primary key nonclustered (ID_POZPARAGON)
)
go
```

```
if exists (select 1 from  sys.extended_properties
           where major_id = object_id('POZPARAGONU') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZPARAGONU'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Pozycja Paragonu',
   'user', @CurrentUser, 'table', 'POZPARAGONU'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZPARAGONU')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_POZPARAGON')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZPARAGONU', 'column', 'ID_POZPARAGON'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer pozycji paragonu.',
   'user', @CurrentUser, 'table', 'POZPARAGONU', 'column', 'ID_POZPARAGON'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZPARAGONU')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_PRODUKTU')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZPARAGONU', 'column', 'ID_PRODUKTU'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer produktu.',
   'user', @CurrentUser, 'table', 'POZPARAGONU', 'column', 'ID_PRODUKTU'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZPARAGONU')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_PARAGONU_')
```

```
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZPARAGONU', 'column', 'ID_PARAGONU_'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer paragonu',
   'user', @CurrentUser, 'table', 'POZPARAGONU', 'column', 'ID_PARAGONU_'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZPARAGONU')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'CENA_NETTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZPARAGONU', 'column', 'CENA_NETTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Cena danego towaru lub us³ugi przed doliczeniem do niej podatku.',
   'user', @CurrentUser, 'table', 'POZPARAGONU', 'column', 'CENA_NETTO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZPARAGONU')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'CENA_BRUTTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZPARAGONU', 'column', 'CENA_BRUTTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Cena towaru lub us³ugi z doliczon¹ kwot¹ podatku VAT.',
   'user', @CurrentUser, 'table', 'POZPARAGONU', 'column', 'CENA_BRUTTO'
go

/*==============================================================*/
/* Index: POZPARAGON_FK                                         */
/*==============================================================*/
create index POZPARAGON_FK on POZPARAGONU (
ID_PARAGONU_ ASC
)
```

```
go

/*==============================================================*/
/* Index: JAKIPRODUKTNAPARAGONIE_FK                             */
/*==============================================================*/
create index JAKIPRODUKTNAPARAGONIE_FK on POZPARAGONU (
ID_PRODUKTU ASC
)
go

/*==============================================================*/
/* Table: POZZAMOWINIA                                          */
/*==============================================================*/
create table POZZAMOWINIA (
   ID_POZZAMOWIENIA     int                 not null,
   ID_PRODUKTU          int                 not null,
   ID_ZAMOWNIA          int                 not null,
   CENA_NETTO           decimal(10,2)       not null,
   CENA_BRUTTO          decimal(10,2)       not null,
   constraint PK_POZZAMOWINIA primary key nonclustered (ID_POZZAMOWIENIA)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('POZZAMOWINIA') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Jest to informacja o poszczególnej pozycji zamówienia wpisanej w system kasowy w
sklepie.',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZZAMOWINIA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_POZZAMOWIENIA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA', 'column', 'ID_POZZAMOWIENIA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer pozycji zamówienia.',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA', 'column', 'ID_POZZAMOWIENIA'
go

if exists(select 1 from sys.extended_properties p where
```

```
      p.major_id = object_id('POZZAMOWINIA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_PRODUKTU')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA', 'column', 'ID_PRODUKTU'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer produktu.',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA', 'column', 'ID_PRODUKTU'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZZAMOWINIA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_ZAMOWNIA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA', 'column', 'ID_ZAMOWNIA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer zamówienia.',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA', 'column', 'ID_ZAMOWNIA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZZAMOWINIA')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'CENA_NETTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA', 'column', 'CENA_NETTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Cena danego towaru lub us³ugi przed doliczeniem do niej podatku.',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA', 'column', 'CENA_NETTO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('POZZAMOWINIA')
```

```
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'CENA_BRUTTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA', 'column', 'CENA_BRUTTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Cena towaru lub us³ugi z doliczon¹ kwot¹ podatku VAT.',
   'user', @CurrentUser, 'table', 'POZZAMOWINIA', 'column', 'CENA_BRUTTO'
go

/*==============================================================*/
/* Index: ELEMENTZAMOWIENIA_FK                                  */
/*==============================================================*/
create index ELEMENTZAMOWIENIA_FK on POZZAMOWINIA (
ID_ZAMOWNIA ASC
)
go

/*==============================================================*/
/* Index: JAKIPRODUKTNAZAMOWIENIU_FK                            */
/*==============================================================*/
create index JAKIPRODUKTNAZAMOWIENIU_FK on POZZAMOWINIA (
ID_PRODUKTU ASC
)
go

/*==============================================================*/
/* Table: PRACOWNIK                                             */
/*==============================================================*/
create table PRACOWNIK (
   ID_PRACOWNIKA        int                  not null,
   ID_DANE              int                  not null,
   IMIE                 varchar(40)          not null,
   NAZWISKO             varchar(40)          not null,
   PESEL                char(11)             not null,
   ROK_URODZENIA        int                  not null,
   STANOWISKO           varchar(20)          not null,
   constraint PK_PRACOWNIK primary key nonclustered (ID_PRACOWNIKA)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('PRACOWNIK') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRACOWNIK'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
```

```
   'Pracownik jest cz³owiekiem zatrudnionym w firmie o okreœlonym stanowisku ',
   'user', @CurrentUser, 'table', 'PRACOWNIK'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRACOWNIK')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_PRACOWNIKA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'ID_PRACOWNIKA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer pracownika.',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'ID_PRACOWNIKA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRACOWNIK')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_DANE')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'ID_DANE'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer danych.',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'ID_DANE'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRACOWNIK')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'IMIE')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'IMIE'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Osobista nazwa nadawana osobie przez grupê, do której nale¿y.',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'IMIE'
```

```
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRACOWNIK')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'NAZWISKO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'NAZWISKO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Nazwa rodziny, do której dana osoba nale¿y',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'NAZWISKO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRACOWNIK')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'PESEL')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'PESEL'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Jedenastocyfrowy symbol numeryczny, który pozwala na ³atw¹ identyfikacjê osoby,
która go posiada.',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'PESEL'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRACOWNIK')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ROK_URODZENIA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'ROK_URODZENIA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Jest to rok przyjœcia na œwiat,',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'ROK_URODZENIA'
go
```

```
if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRACOWNIK')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'STANOWISKO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'STANOWISKO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Informacja na jakiej pozycji pracuje pracownik i jakie mo¿e mieæ przy tym
obowi¹zki.',
   'user', @CurrentUser, 'table', 'PRACOWNIK', 'column', 'STANOWISKO'
go

/*==============================================================*/
/* Index: DANEPRACOWNIKA_FK                                     */
/*==============================================================*/
create index DANEPRACOWNIKA_FK on PRACOWNIK (
ID_DANE ASC
)
go

/*==============================================================*/
/* Table: PRODUKT                                               */
/*==============================================================*/
create table PRODUKT (
   ID_PRODUKTU          int                  not null,
   ID_MARKI             int                  null,
   ID_KOLEKCJI          int                  null,
   MODEL                varchar(50)          null,
   KOLOR                varchar(50)          not null,
   ROZMIAR              int                  not null,
   NR_SERYJNY           char(10)             not null,
   CZY_DOSTEPNY_        char(10)             not null default 'TAK',
   CENA_NETTO           decimal(10,2)        not null,
   RABAT                decimal(2,2)         not null default 0
      constraint CKC_RABAT_PRODUKT check (RABAT between 0 and 1),
   constraint PK_PRODUKT primary key nonclustered (ID_PRODUKTU)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('PRODUKT') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
```

```
   'Produkatmi s¹ buty (Sportowe/Sneakers) dostêpne w sklepie. Ka¿dy produkt
posiada: Cene, Rozmiar, Kolor, inforamcjie czy jest dostêpny oraz mo¿liwy rabat',
   'user', @CurrentUser, 'table', 'PRODUKT'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRODUKT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_PRODUKTU')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'ID_PRODUKTU'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer produktu.',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'ID_PRODUKTU'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRODUKT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_MARKI')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'ID_MARKI'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer marki.',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'ID_MARKI'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRODUKT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_KOLEKCJI')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'ID_KOLEKCJI'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer kolekcji.',
```

```
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'ID_KOLEKCJI'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRODUKT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'MODEL')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'MODEL'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Nazwa modelu z jakiej pochodzi dany produkt.',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'MODEL'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRODUKT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'KOLOR')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'KOLOR'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Odbite œwiat³o, widzimy je i czujemy poprzez oczy, zmys³y i mózg.',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'KOLOR'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRODUKT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ROZMIAR')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'ROZMIAR'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Znormalizowany system okreœlaj¹cy i dopasowuj¹cy pod wzglêdem wielkoœci but i
stopê.',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'ROZMIAR'
```

```
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRODUKT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'NR_SERYJNY')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'NR_SERYJNY'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Kod z³o¿ony z szeregu cyfr lub liter, nadawany produktowi lub serii produktów, w
celu identyfikacji miejsca i czasu wyprodukowania, ustalenia legalnoœci jego
pochodzenia.',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'NR_SERYJNY'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRODUKT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'CZY_DOSTEPNY_')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'CZY_DOSTEPNY_'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Informacja o tym, czy produkt jest dostêpny w sklepie: TAK, NIE oraz SPRZEDANY
(oznaczaj¹cy, ¿e produkt ju¿ nie jest w sklepie a np. u nabywcy)',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'CZY_DOSTEPNY_'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRODUKT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'CENA_NETTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'CENA_NETTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Cena danego towaru lub us³ugi przed doliczeniem do niej podatku.',
```

```
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'CENA_NETTO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('PRODUKT')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'RABAT')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'RABAT'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Informacja o tym o ile zostanie obni¿ona cena produktu.',
   'user', @CurrentUser, 'table', 'PRODUKT', 'column', 'RABAT'
go

/*==============================================================*/
/* Index: JAKAKOLEKCJA_FK                                       */
/*==============================================================*/
create index JAKAKOLEKCJA_FK on PRODUKT (
ID_KOLEKCJI ASC
)
go

/*==============================================================*/
/* Index: JAKAMARKA_FK                                          */
/*==============================================================*/
create index JAKAMARKA_FK on PRODUKT (
ID_MARKI ASC
)
go

/*==============================================================*/
/* Index: JAKIMODEL_FK                                          */
/*==============================================================*/
create index JAKIMODEL_FK on PRODUKT (
MODEL ASC
)
go

/*==============================================================*/
/* Table: ZAMOWIENIE                                            */
/*==============================================================*/
create table ZAMOWIENIE (
   ID_ZAMOWNIA          int                  not null,
   ID_KLIENTA           int                  null,
   RODZAJ_PLATNOSCI     char(10)             not null
      constraint CKC_RODZAJ_PLATNOSCI_ZAMOWIEN check (RODZAJ_PLATNOSCI in
('Karta','Gotowka')),
   KOSZT_NETTO          decimal(10,2)        not null,
   KOSZT_BRUTTO         decimal(10,2)        not null,
   CZY_ZREALZOWANO      char(10)             not null
      constraint CKC_CZY_ZREALZOWANO_ZAMOWIEN check (CZY_ZREALZOWANO in
('ANULOWANO','NIE','TAK') and CZY_ZREALZOWANO = upper(CZY_ZREALZOWANO)),
   DATA                 datetime             not null,
```

```
   constraint PK_ZAMOWIENIE primary key nonclustered (ID_ZAMOWNIA)
)
go

if exists (select 1 from  sys.extended_properties
           where major_id = object_id('ZAMOWIENIE') and minor_id = 0)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Zamówienie jest wpisywane do systemu kas w sklepie. Zawiera informacjê o
kosztach zakupu oraz o tym, czy transakcja zosta³ a zrealizowana.',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('ZAMOWIENIE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_ZAMOWNIA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'ID_ZAMOWNIA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer zamówienia.',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'ID_ZAMOWNIA'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('ZAMOWIENIE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'ID_KLIENTA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'ID_KLIENTA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Unikalny numer klienta sklepu.',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'ID_KLIENTA'
go
```

```
if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('ZAMOWIENIE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'RODZAJ_PLATNOSCI')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'RODZAJ_PLATNOSCI'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Informacja o tym czym zosta³ dokonany zakup: Gotówk¹ lu Kart¹.',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'RODZAJ_PLATNOSCI'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('ZAMOWIENIE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'KOSZT_NETTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'KOSZT_NETTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Koszt danego towaru lub us³ugi przed doliczeniem do niej podatku.',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'KOSZT_NETTO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('ZAMOWIENIE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'KOSZT_BRUTTO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'KOSZT_BRUTTO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Koszt towaru lub us³ugi z doliczon¹ kwot¹ podatku VAT.',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'KOSZT_BRUTTO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('ZAMOWIENIE')
```

```
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'CZY_ZREALZOWANO')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'CZY_ZREALZOWANO'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Informacja o tym, czy zamówienie zosta³o dokonane: TAK, NIE, ANULOWANE
(oznaczaj¹ce ¿e zamówienie nie jest dalej rozpatrywane)',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'CZY_ZREALZOWANO'
go

if exists(select 1 from sys.extended_properties p where
      p.major_id = object_id('ZAMOWIENIE')
  and p.minor_id = (select c.column_id from sys.columns c where c.object_id =
p.major_id and c.name = 'DATA')
)
begin
   declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_dropextendedproperty 'MS_Description',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'DATA'

end


select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
   'Data dokonanego zamówienia.',
   'user', @CurrentUser, 'table', 'ZAMOWIENIE', 'column', 'DATA'
go

/*==============================================================*/
/* Index: ZAMOWIENIE_FK                                         */
/*==============================================================*/
create index ZAMOWIENIE_FK on ZAMOWIENIE (
ID_KLIENTA ASC
)
go

alter table FAKTURA
   add constraint FK_FAKTURA_FAKTURAKU_KLIENT foreign key (ID_KLIENTA)
      references KLIENT (ID_KLIENTA)
go

alter table KLIENT
   add constraint FK_KLIENT_DANEKLIEN_DANE foreign key (ID_DANE)
      references DANE (ID_DANE)
go

alter table PARAGON
   add constraint FK_PARAGON_PARAGONKU_ZAMOWIEN foreign key (ID_ZAMOWNIA)
      references ZAMOWIENIE (ID_ZAMOWNIA)
go
```

```
alter table POZFAKTURY
   add constraint FK_POZFAKTU_JAKIPRODU_PRODUKT foreign key (ID_PRODUKTU)
      references PRODUKT (ID_PRODUKTU)
go

alter table POZFAKTURY
   add constraint FK_POZFAKTU_POZFAKTUR_FAKTURA foreign key (ID_FAKTURA)
      references FAKTURA (ID_FAKTURA)
go

alter table POZPARAGONU
   add constraint FK_POZPARAG_JAKIPRODU_PRODUKT foreign key (ID_PRODUKTU)
      references PRODUKT (ID_PRODUKTU)
go

alter table POZPARAGONU
   add constraint FK_POZPARAG_POZPARAGO_PARAGON foreign key (ID_PARAGONU_)
      references PARAGON (ID_PARAGONU_)
go

alter table POZZAMOWINIA
   add constraint FK_POZZAMOW_ELEMENTZA_ZAMOWIEN foreign key (ID_ZAMOWNIA)
      references ZAMOWIENIE (ID_ZAMOWNIA)
go

alter table POZZAMOWINIA
   add constraint FK_POZZAMOW_JAKIPRODU_PRODUKT foreign key (ID_PRODUKTU)
      references PRODUKT (ID_PRODUKTU)
go

alter table PRACOWNIK
   add constraint FK_PRACOWNI_DANEPRACO_DANE foreign key (ID_DANE)
      references DANE (ID_DANE)
go

alter table PRODUKT
   add constraint FK_PRODUKT_JAKAKOLEK_KOLEKCJA foreign key (ID_KOLEKCJI)
      references KOLEKCJA (ID_KOLEKCJI)
go

alter table PRODUKT
   add constraint FK_PRODUKT_JAKAMARKA_MARKA foreign key (ID_MARKI)
      references MARKA (ID_MARKI)
go

alter table PRODUKT
   add constraint FK_PRODUKT_JAKIMODEL_MODEL foreign key (MODEL)
      references MODEL (MODEL)
go

alter table ZAMOWIENIE
   add constraint FK_ZAMOWIEN_ZAMOWIENI_KLIENT foreign key (ID_KLIENTA)
      references KLIENT (ID_KLIENTA)
go
```

Opisy

Widoki

ASORTYMENT

```
create view [dbo].[ASORTYMENT]
as
	select NAZWA_MARKI
			,Model
			,cast(CENA_NETTO*1.23 as decimal(10,2)) as Cena_Detaliczna
			,CENA_NETTO as Cena_Netto, COUNT(Model) as Ilosc,RABAT as Rabat
			,cast(SUM(Cena_netto)*1.23 as decimal(10,2))as
Wartosc_Sumaryczna
	from PRODUKT, MARKA
	where CZY_DOSTEPNY_ ='TAK'
		and MARKA.ID_MARKI=PRODUKT.ID_MARKI
	group by MARKA.ID_MARKI
			,MODEL
			,CENA_NETTO
			,NAZWA_MARKI
			,RABAT
GO
```

Widok prezentuje zestawienie ile jest dostępnych modeli butów danej marki z danego roku, ile zostało sztuk, jaka jest ich cena netto oraz brutto za sztukę, jaki jest, o ile jest, rabat na daną parę butów oraz jaka jest ich łączna wartość netto.

Rezultat:

	NAZWA_MARKI	Model	Cena_Detaliczna	Cena_Netto	Ilosc	Rabat	Wartosc_Sumaryczna
1	Nike	Air_Force_1	420.00	341.46	25	0.00	10499.90
2	Nike	Air_Force_GTX	500.18	406.65	17	0.00	8503.05
3	Nike	Air_Jordan_1	599.97	487.78	18	0.10	10799.45
4	Nike	Air_Max_720	720.01	585.37	20	0.10	14400.10
5	Nike	Air_Max_95	700.01	569.11	23	0.00	16100.12
6	Adidas	Gazzele	350.00	284.55	20	0.20	6999.93
7	Adidas	Stan_Smith	420.00	341.46	20	0.30	8399.92
8	Adidas	Superstar_Boost	500.18	406.65	24	0.00	12004.31
9	Adidas	Terrex_Swift_R2	519.99	422.76	28	0.00	14559.85
10	Adidas	Ultraboost	750.00	609.76	24	0.00	18000.12
11	Puma	Desierto_V2	400.39	325.52	25	0.00	10009.74
12	Puma	Future_Rider	303.99	247.15	27	0.00	8207.85
13	Puma	RS-X_Luxe	365.49	297.15	23	0.00	8406.37
14	Puma	X-Ray_Lite	365.49	297.15	23	0.00	8406.37
15	Puma	X-Ray_Lite	365.49	297.15	1	0.40	365.49
16	NewBalance	M1500NBR	876.46	712.57	26	0.20	22787.99
17	NewBalance	ML703CLD	360.00	292.68	24	0.00	8639.91
18	NewBalance	Niobium_&_Snow_Peak	900.00	731.71	28	0.00	25200.09
19	NewBalance	UL720UE	299.37	243.39	21	0.00	6286.76
20	Vans	ComfyCush_Slip-On	364.60	296.42	15	0.00	5468.95
21	Vans	Flame_Old_Skool	364.60	296.42	7	0.00	2552.18
22	Vans	Flame_Sk8-hi_Reissue	400.39	325.52	7	0.00	2802.73
23	Vans	Old_Skool_Pro	340.00	276.42	9	0.00	3059.97
24	Vans	Primary_Check_Old_S…	340.00	276.42	14	0.50	4759.95
25	Vans	Sk8-Hi	375.79	305.52	17	0.00	6388.42
26	Vans	Sk8-hi_Mte_2.0_Dx	599.97	487.78	1	0.70	599.97
27	Timberland	Brooklyn_Hiker	370.00	300.81	1	0.20	370.00
28	Timberland	Brooklyn_Hiker	400.39	325.52	12	0.00	4804.68
29	Timberland	Brooklyn_Hiker	400.39	325.52	7	0.20	2802.73
30	Timberland	Premium_6_In	750.00	609.76	17	0.00	12750.08
31	Timberland	Premium_6_In	750.00	609.76	4	0.20	3000.02
32	Timberland	Radford_6_In	800.00	650.41	12	0.00	9600.05
33	Converse	One_Star	370.00	300.81	11	0.20	4069.96
34	Converse	One_Star	370.00	300.81	13	0.30	4809.95

Faktury_Paragony

```
CREATE view [dbo].[Faktury_Parogony]
as
       select ID_FAKTURA as ID_POKWITOWANIA,'Faktura' as
RODZAJ_POKIWTOWANIA,WARTOSC_NETTO,WARTOSC_BRUTTO
              from FAKTURA
union
       select (ID_PARAGONU_) as ID_POKWITOWANIA,'Paragon' as
RODZAJ_POKIWTOWANIA,WARTOSC_NETTO,WARTOSC_BRUTTO
              from PARAGON
GO
```

Widok zawiera zestawienie zawierające informacje o wystawionych fakturach oraz paragonach.

Rezultat:

	ID_POKWITOWANIA	RODZAJ_POKIWTO…	WARTOSC_NETTO	WARTOSC_BRUTTO
1	5	Faktura	868.29	1068.00
2	6	Faktura	138.21	170.00
3	1	Paragon	341.46	420.00
4	2	Paragon	439.00	539.97
5	3	Paragon	341.46	420.00
6	4	Paragon	341.46	420.00

KolekcjeInfo

```
CREATE VIEW [dbo].[KolekcjeInfo]
AS
	SELECT K.KOLEKCJA
		,K.ROK
		,COUNT(DISTINCT M.NAZWA_MARKI) AS IleMarek
		,COUNT(DISTINCT P.MODEL) AS IleModeli
		,COUNT(M.NAZWA_MARKI) AS IleProduktow
	FROM PRODUKT P
			JOIN MARKA M ON p.ID_MARKI = M.ID_MARKI
			JOIN KOLEKCJA K ON K.ID_KOLEKCJI = P.ID_KOLEKCJI
	GROUP BY k.ID_KOLEKCJI
			,K.KOLEKCJA
			,K.ROK
GO
```

Widok prezentuje ile jest marek, modeli oraz produktów z danej kolekcji

Rezultat:

	KOLEKCJA	ROK	IleMarek	IleModeli	IleProduktow
1	Zima	2018	1	1	1
2	Wiosna	2019	1	1	16
3	Wiosna	2020	6	7	152
4	Lato	2020	5	6	125
5	Jesien	2020	5	6	121
6	Zima	2020	5	7	131
7	Zima	2021	3	4	78

Procedury

W tej bazie danych są trzy procedury umożliwiające dodawanie zamówień oraz wystawianie faktur oraz paragonów.

DodajDoZamowienia

```
CREATE
 PROCEDURE [dbo].[DodajDoZamowienia]
       @Id_Zamowienia INT
       ,@Id_Produktu INT
       ,@Rodzaj_plat CHAR(10) = 'Gotowka'
       ,@Id_Klienta INT = NULL
AS
IF (
              SELECT CZY_ZREALZOWANO
              FROM ZAMOWIENIE
              WHERE ID_ZAMOWNIA = @Id_Zamowienia
              ) = 'NIE' OR (
              SELECT CZY_ZREALZOWANO
              FROM ZAMOWIENIE
              WHERE ID_ZAMOWNIA = @Id_Zamowienia
              ) IS NULL
BEGIN
       BEGIN TRY
              BEGIN TRAN

              --select * from POZZAMOWINIA
              IF NOT EXISTS (
                           SELECT 1
                           FROM ZAMOWIENIE
                           WHERE ID_ZAMOWNIA = @Id_Zamowienia
                           )
              BEGIN
                     INSERT INTO ZAMOWIENIE (
                           ID_ZAMOWNIA
                           ,ID_KLIENTA
                           ,RODZAJ_PLATNOSCI
                           ,KOSZT_NETTO
                           ,KOSZT_BRUTTO
                           ,CZY_ZREALZOWANO
                           ,DATA
                           )
                     VALUES (
                           @Id_Zamowienia
                           ,@Id_Klienta
                           ,@Rodzaj_plat
                           ,0.0
                           ,0.0
                           ,'NIE'
                           ,getdate()
                           )
              END

              DECLARE @Id_Poz INT
              DECLARE @CN DECIMAL(10, 2)

              SET @CN = ([dbo].[LiczPromo](@Id_Produktu))

              IF (
                           (
```

```
                              SELECT CZY_DOSTEPNY_
                              FROM PRODUKT
                              WHERE ID_PRODUKTU = @Id_Produktu
                              ) = 'TAK'
                       )
       BEGIN
              INSERT INTO POZZAMOWINIA (
                     ID_ZAMOWNIA
                     ,ID_PRODUKTU
                     ,CENA_NETTO
                     ,CENA_BRUTTO
                     )
              VALUES (
                     @Id_Zamowienia
                     ,@Id_Produktu
                     ,@CN
                     ,(
                              SELECT [dbo].[LiczBrutto](@CN, 1.23)
                              )
                     )

              UPDATE ZAMOWIENIE
              SET KOSZT_NETTO = (
                              SELECT SUM(CENA_NETTO)
                              FROM POZZAMOWINIA
                              WHERE ID_ZAMOWNIA = @Id_Zamowienia
                              )
              WHERE ID_ZAMOWNIA = @Id_Zamowienia

              UPDATE ZAMOWIENIE
              SET KOSZT_BRUTTO = (
                              SELECT SUM(CENA_BRUTTO)
                              FROM POZZAMOWINIA
                              WHERE ID_ZAMOWNIA = @Id_Zamowienia
                              )
              WHERE ID_ZAMOWNIA = @Id_Zamowienia
       END

       IF (
                       (
                              SELECT CZY_DOSTEPNY_
                              FROM PRODUKT
                              WHERE ID_PRODUKTU = @Id_Produktu
                              ) = 'TAK'
                       )
       BEGIN
              UPDATE PRODUKT
              SET CZY_DOSTEPNY_ = 'NIE'
              WHERE ID_PRODUKTU = @Id_Produktu
       END
       ELSE
       BEGIN
              PRINT ('Produkt niedostepny')
       END

       IF @@TRANCOUNT > 0
              COMMIT TRAN
END TRY

BEGIN CATCH
       SELECT ERROR_NUMBER() AS ErrorNumber
              ,ERROR_SEVERITY() AS ErrorSeverity
```

```
                    ,ERROR_STATE() AS ErrorState
                    ,ERROR_PROCEDURE() AS ErrorProcedure
                    ,ERROR_LINE() AS ErrorLine
                    ,ERROR_MESSAGE() AS ErrorMessage

            IF @@TRANCOUNT > 0
                    ROLLBACK TRANSACTION;
      END CATCH
END
ELSE
BEGIN
      PRINT 'Zamowienie zostalo zrealizowane'
END
GO
```

Sposób użycia:
Procedura ta wykorzystuje mechanizm transakcji oraz przyjmuje cztery argumenty: Id_Zamowienia, Id_Produktu, Rodzaj_plat, Id_Klienta. Jako wynik działania tworzy nowe zamówienie w tabeli ZAMOWIENIA, jeśli zamówienie nie istnieje oraz dzięki podanym argumentom tworzy kolejne wiersze w tabeli POZZAMOWIENIA gdzie przechowywane są informacje o tym jaki produkt znajduje się na zamówieniu. Warto zauważyć, że każdy produkt w tabeli produkty posiada informacje o tym czy jest dostępny lub czy został sprzedany. Po dodaniu produktu do zamówienia jego status jest zmieniany na niedostępny co nie pozwala na ponowne dodanie tego produktu do innego zamówienia. Na sam koniec aktualizowana jest zawartość tabeli zamówienie, gdzie trafia informacja o należności jaką należałoby zapłacić w przypadku realizacji zamówienia. Podobnie jak tabela produkt, tabela zamówienie posiada informacje o tym czy zamówienie zostało zrealizowane czy też nie. Do zamówienia można dodawać elementy, dopóki te nie jest zrealizowane.

Dodamy do zamówienia dwa produkty o ID_PRODUKTU = 1 oraz ID_PRODUKTU = 2:

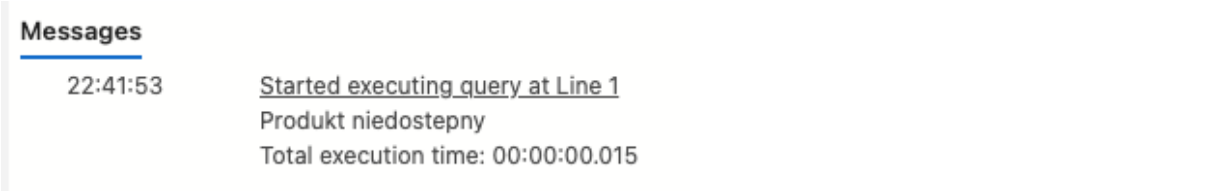
	ID_PRODUKTU	ID_KOLEKCJI	MODEL	ID_MARKI	KOLOR	ROZMIAR	NR_SERYJNY	CZY_DOSTEPNY_	CENA_NETTO	RABAT
1	1	9	Air_Force_1	1	Niebieski	38	123456789	NIE	341.46	0.00
2	2	9	Air_Force_1	1	Czarny	45	JZTV70DT1O	TAK	341.46	0.00

```
DECLARE @RC int
DECLARE @Id_Zamowienia int
DECLARE @Id_Produktu int
DECLARE @Rodzaj_plat char(10)
DECLARE @Id_Klienta int

-- TODO: Set parameter values here.

EXECUTE @RC = [dbo].[DodajDoZamowienia]
@Id_Zamowienia=3
,@Id_Produktu=1
,@Rodzaj_plat='Gotowka'
,@Id_Klienta=NULL
GO
```

otrzymujemy wynik:



Czyli widzimy rezultat jakim jest informacja o tym, że produkt nie jest dostępny co jest zgodne z prawdą, ponieważ ID_PRODUKTU = 1 ma CZY_DOSTEPNY_ = NIE.

Teraz drugi przykład z dodaniem właściwego produktu do zamówienia:

```
DECLARE @RC int
DECLARE @Id_Zamowienia int
DECLARE @Id_Produktu int
DECLARE @Rodzaj_plat char(10)
DECLARE @Id_Klienta int

-- TODO: Set parameter values here.

EXECUTE @RC = [dbo].[DodajDoZamowienia]
@Id_Zamowienia=3
,@Id_Produktu=2
,@Rodzaj_plat='Gotowka'
,@Id_Klienta=NULL
GO
```

Patrząc do tabeli PRODUKT możemy zauważyć, że nastąpiła zmiana CZY_DOSTEPNY wybranego produktu:

	ID_PRODUKTU	ID_KOLEKCJI	MODEL	ID_MARKI	KOLOR	ROZMIAR	NR_SERYJNY	CZY_DOSTEPNY_	CENA_NETTO	RABAT
1	2	9	Air_Force_1	1	Czarny	45	JZTV70DT10	NIE	341.46	0.00

oraz:

	ID_ZAMOWNIA	ID_KLIENTA	RODZAJ_PLATNOSCI	KOSZT_NETTO	KOSZT_BRUTTO	CZY_ZREALZOWANO	DATA
1	3	*NULL*	Gotowka	341.46	420.00	NIE	2022-01-26

	ID_POZZAMOWIENIA	ID_ZAMOWNIA	ID_PRODUKTU	CENA_NETTO	CENA_BRUTTO
1	3	3	2	341.46	420.00

Kolejny produkt dodany do zamówienia:

	ID_ZAMOWNIA	ID_KLIENTA	RODZAJ_PLATNOSCI	KOSZT_NETTO	KOSZT_BRUTTO	CZY_ZREALZOWANO	DATA
1	3	*NULL*	Gotowka	868.29	1068.00	NIE	2022-01-26

	ID_POZZAMOWIENIA	ID_ZAMOWNIA	ID_PRODUKTU	CENA_NETTO	CENA_BRUTTO
1	3	3	2	341.46	420.00
2	4	3	50	526.83	648.00

DodajDoFaktury

```
CREATE
 PROCEDURE [dbo].[DodajDoFaktury] @Id_Zamowienia INT
       ,@Id_Klienta INT = NULL
       ,@NIP DECIMAL(10, 0)
       ,@NazwaBanku VARCHAR(50) = NULL
AS
IF (
              SELECT CZY_ZREALZOWANO
              FROM ZAMOWIENIE
              WHERE ID_ZAMOWNIA = @Id_Zamowienia
              ) = 'NIE'
BEGIN
       IF (
                     SELECT RODZAJ_PLATNOSCI
                     FROM ZAMOWIENIE
                     WHERE ID_ZAMOWNIA = @Id_Zamowienia
                     ) = 'Gotowka'
       BEGIN
              SET @NazwaBanku = NULL
       END

       BEGIN TRY
              BEGIN TRAN

              DECLARE @Id_P INT
              DECLARE @C_B DECIMAL(10, 2)
              DECLARE @C_N DECIMAL(10, 2)
              DECLARE @Id_PozF INT
              DECLARE @Id_F INT

              INSERT INTO FAKTURA (
                     ID_KLIENTA
                     ,WARTOSC_NETTO
                     ,WARTOSC_BRUTTO
                     ,WARTOSC_VAT
                     ,NAZWA_BANKU
                     ,RODZAJ_PLATNOSCI
                     ,NIP
                     )
              VALUES (
                     @Id_Klienta
                     ,0.0
                     ,0.0
                     ,0.0
                     ,@NazwaBanku
                     ,(
                            SELECT RODZAJ_PLATNOSCI
                            FROM ZAMOWIENIE
                            WHERE ID_ZAMOWNIA = @Id_Zamowienia
                            )
                     ,@NIP
                     )

              SET @Id_F=(SELECT MAX(ID_FAKTURA) FROM  FAKTURA)

              IF (
                            SELECT ID_KLIENTA
                            FROM ZAMOWIENIE
                            WHERE ID_ZAMOWNIA = @Id_Zamowienia
                            ) IS NULL
```

```
BEGIN
	UPDATE ZAMOWIENIE
	SET ID_KLIENTA = @Id_Klienta
	WHERE ID_ZAMOWNIA = @Id_Zamowienia
END

DECLARE KURSOR CURSOR
FOR
SELECT ID_PRODUKTU
	,CENA_BRUTTO
	,CENA_NETTO
FROM POZZAMOWINIA POZ
WHERE POZ.ID_ZAMOWNIA = @Id_Zamowienia

OPEN KURSOR

FETCH NEXT
FROM KURSOR
INTO @Id_P
	,@C_B
	,@C_N

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

WHILE @@FETCH_STATUS = 0
BEGIN
	IF (
			(
				SELECT MAX(POZF.ID_POZFAKTURY)
				FROM POZFAKTURY POZF
				) IS NULL
			)
	BEGIN
		SET @Id_PozF = 1
	END
	ELSE
	BEGIN
		SET @Id_PozF = (
				SELECT MAX(POZF.ID_POZFAKTURY) + 1
				FROM POZFAKTURY POZF
				)
	END

	INSERT INTO POZFAKTURY (
		ID_FAKTURA
		,ID_PRODUKTU
		,CENA_BRUTTO
		,CENA_NETTO
		)
	VALUES (
		@Id_F
		,@Id_P
		,@C_B
		,@C_N
		)

	FETCH NEXT
	FROM KURSOR
	INTO @Id_P
		,@C_B
		,@C_N
```

```
		END

		CLOSE KURSOR

		DEALLOCATE KURSOR

		DECLARE @SumaCN DECIMAL(10, 2)

		SET @SumaCN = (
				SELECT sum(CENA_NETTO)
				FROM POZFAKTURY
				WHERE ID_FAKTURA = @Id_F
				)

		UPDATE FAKTURA
		SET WARTOSC_BRUTTO = dbo.LiczBrutto(@SumaCN, 1.23)
			,WARTOSC_NETTO = @SumaCN
			,WARTOSC_VAT = dbo.LiczVAT(@SumaCN)
		WHERE ID_FAKTURA = @Id_F

		UPDATE ZAMOWIENIE
		SET CZY_ZREALZOWANO = 'TAK'
		WHERE ID_ZAMOWNIA = @Id_Zamowienia

		IF @@TRANCOUNT > 0
			COMMIT TRAN
	END TRY

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

		IF @@TRANCOUNT > 0
			ROLLBACK TRANSACTION;
	END CATCH
END
ELSE
BEGIN
	PRINT 'Juz wystawiono fakture'
END
GO
```

Sposób użycia:

Procedura wykorzystuje mechanizm transakcji, kursora oraz przyjmuje cztery argumenty: Id_Zamowienia, Id_Klienta, NIP, NazwaBanku. Procedura ta tworzy nową fakturę z podanym przez użytkownika NIP-em oraz nazwą Banku, ale w przypadku gdy zamówienie jest opłacone gotówką to parametr ten nie jest konieczny i jest ustawiany na wartość NULL. Za pomocą przyjmowanych argumentów przepisywana jest wartość za pomocą kursora z tabeli POZZAMOWIENIA o wskazanym przez użytkownika Id_Zamowienia do tabeli POZFAKTURY. Po dodaniu nowej faktury status zamówienia jest zmieniany na zrealizowany.

Próba wystawienia faktury na zamówienie 3:

```
DECLARE @RC int
DECLARE @Id_Zamowienia int
DECLARE @Id_Klienta int
DECLARE @NIP decimal(10,0)
DECLARE @NazwaBanku varchar(50)

-- TODO: Set parameter values here.

EXECUTE @RC = [dbo].[DodajDoFaktury]
@Id_Zamowienia=3
,@Id_Klienta=1
,@NIP=1122334455
,@NazwaBanku=NULL
GO
```

Rezultat:

	ID_FAKTURA	ID_KLIENTA	WARTOSC_NETTO	WARTOSC_BRUTTO	WARTOSC_VAT	NAZWA_BANKU	RODZAJ_PLATNOSCI	NIP
1	5	1	868.29	1068.00	199.71	*NULL*	Gotowka	1122334455

	ID_POZFAKTURY	ID_FAKTURA	ID_PRODUKTU	CENA_NETTO	CENA_BRUTTO
1	3	5	2	341.46	420.00
2	4	5	50	526.83	648.00

	ID_ZAMOWNIA	ID_KLIENTA	RODZAJ_PLATNOSCI	KOSZT_NETTO	KOSZT_BRUTTO	CZY_ZREALZOWANO	DATA
1	3	1	Gotowka	868.29	1068.00	TAK	2022-01-26 22:40:36.920

Widać że status zamówienia został zmieniony na zrealizowany oraz została stworzona nowa faktura, na którą trafiły odpowiednie produkty.

DodajDoParagonu

```
CREATE
 PROCEDURE [dbo].[DodajDoParagonu] @Id_Zamowienia INT
       ,@Id_Klienta INT = NULL
AS
IF (
              SELECT CZY_ZREALZOWANO
              FROM ZAMOWIENIE
              WHERE ID_ZAMOWNIA = @Id_Zamowienia
              ) = 'NIE'
BEGIN
       BEGIN TRY
              BEGIN TRAN

              DECLARE @Id_P INT
              DECLARE @C_B DECIMAL(10, 2)
              DECLARE @C_N DECIMAL(10, 2)
              DECLARE @Id_PozPa INT
              DECLARE @Id_Pa INT




              INSERT INTO PARAGON (
                     ID_ZAMOWNIA
                     ,WARTOSC_BRUTTO
                     ,WARTOSC_NETTO
                     ,RODZAJ_PLATNOSCI
                     )
              VALUES (
                     @Id_Zamowienia
                     ,0.0
                     ,0.0
                     ,(
                           SELECT RODZAJ_PLATNOSCI
                           FROM ZAMOWIENIE
                           WHERE ID_ZAMOWNIA = @Id_Zamowienia
                           )
                     )


              SET @Id_Pa = (SELECT MAX(ID_PARAGONU_) FROM PARAGON)


              IF (
                           SELECT ID_KLIENTA
                           FROM ZAMOWIENIE
                           WHERE ID_ZAMOWNIA = @Id_Zamowienia
                           ) IS NULL
              BEGIN
                     UPDATE ZAMOWIENIE
                     SET ID_KLIENTA = @Id_Klienta
                     WHERE ID_ZAMOWNIA = @Id_Zamowienia
              END

              DECLARE KURSOR3 CURSOR
              FOR
              SELECT ID_PRODUKTU
                     ,CENA_NETTO
                     ,CENA_BRUTTO
              FROM POZZAMOWINIA POZ
```

```
		WHERE POZ.ID_ZAMOWNIA = @Id_Zamowienia

		OPEN KURSOR3

		FETCH NEXT
		FROM KURSOR3
		INTO @Id_P
			,@C_N
			,@C_B

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

		WHILE @@FETCH_STATUS = 0
		BEGIN

			INSERT INTO POZPARAGONU (
				ID_PARAGONU_
				,ID_PRODUKTU
				,CENA_BRUTTO
				,CENA_NETTO
				)
			VALUES (
				@Id_Pa
				,@Id_P
				,@C_B
				,@C_N
				)


			FETCH NEXT
			FROM KURSOR3
			INTO @Id_P
				,@C_N
				,@C_B
		END

		CLOSE KURSOR3

		DEALLOCATE KURSOR3

		DECLARE @temp DECIMAL(10, 2)

		SET @temp = (
				SELECT sum(CENA_NETTO)
				FROM POZPARAGONU
				WHERE ID_PARAGONU_ = @Id_Pa
				)

		UPDATE PARAGON
		SET WARTOSC_BRUTTO = dbo.LiczBrutto(@temp,1.23)
			,WARTOSC_NETTO = @temp
		WHERE ID_PARAGONU_ = @Id_Pa

		UPDATE ZAMOWIENIE
		SET CZY_ZREALZOWANO = 'TAK'
		WHERE ID_ZAMOWNIA = @Id_Zamowienia


		IF @@TRANCOUNT > 0
			COMMIT TRAN
	END TRY
```

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

		IF @@TRANCOUNT > 0
			ROLLBACK TRANSACTION;
	END CATCH
END
ELSE
BEGIN
	PRINT 'Juz wystawiono paragon'
END
GO
```

Sposób użycia:
Procedura wykorzystuje mechanizm transakcji, kursora oraz przyjmuje dwa argumenty: Id_Zamowienia, Id_Klienta. Podobnie jak we wcześniejszej procedurze tworzy ona nową pozycję w tabeli PARAGON oraz przepisuje z tabeli POZZAMOWIENIA do tabeli POZPARAGONU oraz uaktualnia status zamówienia na zrealizowany.

Jako, że wystawienie paragonu działa podobnie do wystawienia faktury to sprawdzimy, czy możemy wystawić paragon na zamówienie, na które zostało już wystawione inne pokwitowanie:

```
DECLARE @RC int
DECLARE @Id_Zamowienia int
DECLARE @Id_Klienta int

-- TODO: Set parameter values here.

EXECUTE @RC = [dbo].[DodajDoParagonu]
@Id_Zamowienia=3
,@Id_Klienta=NULL
GO
```

Rezultat:

```
Started executing query at Line 1
Juz wystawiono paragon
Total execution time: 00:00:00.061
```

Prawdą jest, że nie był to paragon, jednak komunikat oznacza, że zamówienie zostało już zrealizowane i nie można wystawić drugiego pokwitowania. Gdyby tak nie było to wynik działania byłby analogiczny do tego z poprzedniej procedury.

Funkcje

LiczPromo

```
CREATE
 FUNCTION [dbo].[LiczPromo] (@Id_Pr INT)
RETURNS DECIMAL(10, 2)
AS
BEGIN
	DECLARE @RabatLicz DECIMAL(10, 2)
	DECLARE @Rabat DECIMAL(10, 2)
	DECLARE @CN DECIMAL(10, 2)

	SET @Rabat = (
			SELECT RABAT
			FROM PRODUKT
			WHERE ID_PRODUKTU = @Id_Pr
			)
	SET @CN = (
			SELECT CENA_NETTO
			FROM PRODUKT
			WHERE ID_PRODUKTU = @Id_Pr
			)

	IF (@Rabat <> 0.0)
	BEGIN
		SET @RabatLicz = 1.0 - @Rabat
	END
	ELSE
	BEGIN
		SET @RabatLicz = 1.0
	END

	RETURN (cast(@CN * @RabatLicz AS DECIMAL(10, 2)))
END
GO
```

Jako argument funkcja przyjmuje Id_produktu a jako wynik wylicza wartość ceny po obniżce dla wskazanego produktu. Natomiast jeżeli produkt nie jest przeceniony zwraca nam zwykłą cenę netto. Działa to w ten sposób, ponieważ rabat na produkt jest wyrażony jako liczba z zakresu od zera do jeden.

Cena produktu o ID_PRODUKTU = 482:

	ID_PRODUKTU	CENA_NETTO	RABAT
1	482	276.42	0.50

Po użyciu funkcji:

	CENA_PO_RABACIE
1	138.21

IleJest

```
CREATE
 FUNCTION [dbo].[IleJest] (
    @Szukam CHAR(10)
       )
RETURNS INT
AS
BEGIN
    DECLARE @Nazwa CHAR(10)
    DECLARE @Liczba INT

    SET @Liczba = 0

    DECLARE kursor CURSOR
    FOR
       SELECT CZY_DOSTEPNY_
       FROM PRODUKT

       OPEN kursor

       FETCH NEXT
       FROM kursor
       INTO @Nazwa

       WHILE @@FETCH_STATUS = 0
       BEGIN
        IF @Nazwa = @Szukam
        BEGIN
            SET @Liczba = @Liczba + 1
        END
    FETCH NEXT
    FROM kursor
    INTO @Nazwa
    END

    CLOSE kursor

    DEALLOCATE kursor

    RETURN @Liczba
END
GO
```

Funkcja przyjmuje jako argument wartość z kolumny CZY_DOSTĘPNY_, czyli „TAK”, „NIE”, „SPRZEDANO” i oblicza ile jest sztuk produktu pożądanej informacji o dostępności.

Rezultat:

	LICZA_PRODUKTOW_Z_TAK
1	554

	LICZA_PRODUKTOW_Z_SPRZEDANY
1	9

	LICZA_PRODUKTOW_Z_NIE
1	61

SredniaMarki

```
CREATE
 FUNCTION [dbo].[SredniaMarki] (
       @Id_Mar INT
       )
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @Srednia DECIMAL(10,2)
    DECLARE @Suma DECIMAL(10,2)
    DECLARE @Ceny DECIMAL(10,2)
    DECLARE @Zmienna INT

    SET @Suma = 0
    SET @Zmienna = 0

    DECLARE kursor CURSOR
    FOR
       SELECT CENA_NETTO
       FROM PRODUKT
    WHERE ID_MARKI = @Id_Mar
    GROUP BY CENA_NETTO

       OPEN kursor

       FETCH NEXT
       FROM kursor
       INTO @Ceny

       WHILE @@FETCH_STATUS = 0
       BEGIN
        SET @Suma = @Suma + @Ceny
        SET @Zmienna = @Zmienna + 1
        FETCH NEXT
        FROM kursor
        INTO @Ceny
    END

    IF @Zmienna <> 0
    BEGIN
        SET @Srednia = @Suma/@Zmienna
    END
    ELSE
    BEGIN
        SET @Srednia = 0
    END

    CLOSE kursor

    DEALLOCATE kursor

    RETURN @Srednia
END
GO
```

Funkcja przyjmuje jako argument Id_Marki i wylicza jaka jest średnia cena butów danej marki

Rezultat:

	SREDNIA_CENY_MARKI
1	478.07

Triggery

FakturaSprzedano

Wyzwalacz ten jest używany po każdym dodaniu elementu do tabeli POZFAKTURY. Jego zadaniem jest zmiana statusu produktu w tabeli PRODUKTY po jego sprzedaniu.

```
CREATE
 TRIGGER [dbo].[FakturaSprzedano] ON [dbo].[POZFAKTURY]
AFTER INSERT
AS
BEGIN
	DECLARE @ID_P INT

	SET @ID_P = (
			SELECT ID_PRODUKTU
			FROM POZFAKTURY
			WHERE ID_POZFAKTURY = (
					SELECT max(ID_POZFAKTURY)
					FROM POZFAKTURY
					)
			)

	UPDATE PRODUKT
	SET CZY_DOSTEPNY_ = 'SPRZEDANY'
	WHERE ID_PRODUKTU = @ID_P
END
GO
ALTER TABLE [dbo].[POZFAKTURY] ENABLE TRIGGER [FakturaSprzedano]
GO
```

Sposób użycia:
Dla wystawionej wcześniej faktury:

	ID_FAKTURA	ID_KLIENTA	WARTOSC_NETTO	WARTOSC_BRUTTO	WARTOSC_VAT	NAZWA_BANKU	RODZAJ_PLATNOSCI	NIP
1	5	1	868.29	1068.00	199.71	*NULL*	Gotowka	1122334455

	ID_POZFAKTURY	ID_FAKTURA	ID_PRODUKTU	CENA_NETTO	CENA_BRUTTO
1	3	5	2	341.46	420.00
2	4	5	50	526.83	648.00

	ID_ZAMOWNIA	ID_KLIENTA	RODZAJ_PLATNOSCI	KOSZT_NETTO	KOSZT_BRUTTO	CZY_ZREALZOWANO	DATA
1	3	1	Gotowka	868.29	1068.00	TAK	2022-01-26 22:40:36.920

Można zauważyć, że na fakturze widać dwa produkty, o ID_PRODUKTU równym dwa oraz pięćdziesiąt. Ich status zmienił się na sprzedany, co można jeszcze zauważyć za pomocą zapytania:

```
select * from PRODUKT where ID_PRODUKTU=2 or ID_PRODUKTU=50
```

	ID_PRODUKTU	ID_KOLEKCJI	MODEL	ID_MARKI	KOLOR	ROZMIAR	NR_SERYJNY	CZY_DOSTEPNY_	CENA_NETTO	RABAT
1	2	9	Air_Force_1	1	Czarny	45	JZTV70DT1O	SPRZEDANY	341.46	0.00
2	50	10	Air_Max_720	1	Czarny	43	ZFKEZLTJFE	SPRZEDANY	585.37	0.10

ParagonSprzedano

```
CREATE
 TRIGGER [dbo].[ParagonSprzedono] ON [dbo].[POZPARAGONU]
AFTER INSERT
AS
BEGIN
	DECLARE @ID_P INT

	SET @ID_P = (
			SELECT ID_PRODUKTU
			FROM POZPARAGONU
			WHERE ID_POZPARAGON = (
					SELECT max(ID_POZPARAGON)
					FROM POZPARAGONU
					)
			)

	UPDATE PRODUKT
	SET CZY_DOSTEPNY_ = 'SPRZEDANY'
	WHERE ID_PRODUKTU = @ID_P
END
GO
ALTER TABLE [dbo].[POZPARAGONU] ENABLE TRIGGER [ParagonSprzedono]
GO
```

Wyzwalacz działa analogicznie do wcześniej opisanego.

AnulowaneZamowienie

```
CREATE

 TRIGGER [dbo].[AnulowaneZamowienie] ON [dbo].[ZAMOWIENIE]
AFTER UPDATE
AS
BEGIN
	IF EXISTS (
			SELECT 1
			FROM ZAMOWIENIE
			WHERE CZY_ZREALZOWANO = 'ANULOWANO'
			)
	BEGIN
		DECLARE @Id_Zamowienia INT
		DECLARE @Id_Produktu INT

		SET @Id_Zamowienia = (
				SELECT TOP 1 ID_ZAMOWNIA
				FROM ZAMOWIENIE Z
				WHERE Z.CZY_ZREALZOWANO = 'ANULOWANO'
				ORDER BY Z.DATA DESC
				)

		DECLARE Coursorek CURSOR
		FOR
		SELECT ID_PRODUKTU
		FROM POZZAMOWINIA
		WHERE ID_ZAMOWNIA = @Id_Zamowienia

		OPEN Coursorek

		FETCH NEXT
		FROM Coursorek
		INTO @Id_Produktu

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

		WHILE @@FETCH_STATUS = 0
		BEGIN
			UPDATE PRODUKT
			SET CZY_DOSTEPNY_ = 'TAK'
			WHERE ID_PRODUKTU = @Id_Produktu


			FETCH NEXT
			FROM Coursorek
			INTO @Id_Produktu
		END

		CLOSE Coursorek

		DEALLOCATE Coursorek
	END --if1
END
GO
ALTER TABLE [dbo].[ZAMOWIENIE] ENABLE TRIGGER [AnulowaneZamowienie]
GO
```

Celem działania tego triggera jest zmiana statusu produktów dodanych do zamówienia w przypadku, gdy te zostanie anulowane. Wyzwalany jest po tym jak status zamówienia zostanie zmieniony na „ANULOWANO”.

Sposób użycia:
Dla zamówienia:

	ID_PRODUKTU	ID_KOLEKCJI	MODEL	ID_MARKI	KOLOR	ROZMIAR	NR_SERYJNY	CZY_DOSTEPNY_	CENA_NETTO	RABAT
1	100	13	Air_Jordan_1	1	Czarny	45	ZN7TLOJDH8	NIE	487.78	0.10
2	200	12	Terrex_Swift_R2	2	Czarny	43	WZFX1COMWQ	NIE	422.76	0.00
3	300	11	RS-X_Luxe	3	Bialy	40	JRHX2SC94W	NIE	297.15	0.00

	ID_ZAMOWNIA	ID_KLIENTA	RODZAJ_PLATNOSCI	KOSZT_NETTO	KOSZT_BRUTTO	CZY_ZREALZOWANO	DATA
1	4	*NULL*	Gotowka	1158.91	1425.45	NIE	2022-01-26 23:55:37.397

	ID_POZZAMOWIENIA	ID_ZAMOWNIA	ID_PRODUKTU	CENA_NETTO	CENA_BRUTTO
1	5	4	100	439.00	539.97
2	6	4	200	422.76	519.99
3	7	4	300	297.15	365.49

Po użyciu UPDATE:

```
UPDATE ZAMOWIENIE
SET CZY_ZREALZOWANO='ANULOWANO'
WHERE ID_ZAMOWNIA=4
```

	ID_PRODUKTU	ID_KOLEKCJI	MODEL	ID_MARKI	KOLOR	ROZMIAR	NR_SERYJNY	CZY_DOSTEPNY_	CENA_NETTO	RABAT
1	100	13	Air_Jordan_1	1	Czarny	45	ZN7TLOJDH8	TAK	487.78	0.10
2	200	12	Terrex_Swift_R2	2	Czarny	43	WZFX1COMWQ	TAK	422.76	0.00
3	300	11	RS-X_Luxe	3	Bialy	40	JRHX2SC94W	TAK	297.15	0.00

	ID_ZAMOWNIA	ID_KLIENTA	RODZAJ_PLATNOSCI	KOSZT_NETTO	KOSZT_BRUTTO	CZY_ZREALZOWANO	DATA
1	4	*NULL*	Gotowka	1158.91	1425.45	ANULOWANO	2022-01-26 23:55:37.397

	ID_POZZAMOWIENIA	ID_ZAMOWNIA	ID_PRODUKTU	CENA_NETTO	CENA_BRUTTO
1	5	4	100	439.00	539.97
2	6	4	200	422.76	519.99
3	7	4	300	297.15	365.49

Użytkownicy

```
CREATE LOGIN [ADMIN] WITH PASSWORD = 'Adminek1!';
go
CREATE USER [SYSADMIN] FOR LOGIN [ADMIN]
EXEC sp_addrolemember 'db_owner', ' SYSADMIN'
Go


CREATE LOGIN [PRACOWNIK] WITH PASSWORD = 'Pracownik1!';
go
CREATE USER [PRACOWNIK] FOR LOGIN [PRACOWNIK]
EXEC sp_addrolemember 'db_datareader', ' PRACOWNIK'
go


CREATE LOGIN [KIEROWNIK] WITH PASSWORD = 'Kierownik1!';
go
CREATE USER [KIEROWNIK] FOR LOGIN [KIERWONIK]
EXEC sp_addrolemember 'db_datareader', ' KIEROWNIK'
EXEC sp_addrolemember 'db_datawriter', ' KIEROWNIK'
```

Wnioski

Możliwości jakie dają wyzwalacze, procedury oraz funkcje są bardzo rozległe i ciężko byłoby wskazać je wszystkie. Uważam, że najważniejsze z nich to to, że pozwalają one usprawnić i zautomatyzować dodawanie danych. Można powiedzieć, że dzięki temu baza danych zaczyna „żyć” i wykonuje niektóre czynności sama bez konieczności interwencji administratora lub innego użytkownika.