

BAZY DANYCH

SPRAWOZDANIE Z PROJEKTU

LABORATORIUM



Temat:

Voltra Energy

*Grupowe zakupy energii elektrycznej oraz paliwa gazowego
dla Jednostek Samorządowych*

Prowadzący: **plk dr inż. Jarosław Koszela**

Autor: *Mariusz Celiński-Mysław*

Spis treści

1. Zagadnienia obejmujące projekt	3
2. Tematyka oraz założenia projektu.....	3
3. Model konceptualny	4
4. Model fizyczny	6
5. Listing kodu generującego bazę danych.....	7
6. Widoki	35
a) Widok funkcyjny - v_punkty_do_zgloszenia	35
b) Widok funkcyjny - v_najlepszy_zespól	37
c) Widok zmaterializowany - vm_wynik_roczny	38
7. Procedury	39
a) Procedura z użyciem kursora - obnizka_ceny	39
b) Procedura z użyciem transakcji - aktualizacja_ppe.....	42
c) Procedura z użyciem transakcji - dodaj_pracownika	44
8. Funkcje	46
a) Funkcja z użyciem kursora - podwyzka	46
b) Funkcja z użyciem kursora - obrot	47
c) Funkcja z użyciem kursora - wartosc_faktury	48
d) Funkcja z użyciem transakcji i kursora - odprawa_pracownika.....	49
e) Funkcja z użyciem kursora - wielkość_wadium	50
f) Funkcja z użyciem kursora - obecna_stawka	51
9. Wyzwalacze (triggery).....	52
a) Wyzwalacz – aktualizacja_wolumenu	52
b) Wyzwalacz - status_wadium.....	53
c) Wyzwalacz – najkorzystniejsza_oferta.....	54
d) Wyzwalacz – zmiana_ceny.....	55

1. Zagadnienia obejmujące projekt

Stworzenie modelu conceptualnego oraz fizycznego bazy danych. Projekt ma zawierać:

- Co najmniej 3 widoki
 - Co najmniej 1 widok funkcyjny
 - Co najmniej 1 widok zmaterializowany
- Co najmniej 3 procedury
 - Co najmniej 1 procedura zawierająca transakcje
 - Co najmniej 1 procedura zawierająca kursor
- Co najmniej 3 funkcje
 - Co najmniej 1 funkcja zawierająca transakcje
 - Co najmniej 1 funkcja zawierająca kursor
- Co najmniej 3 wyzwalacze (triggery)
- Co najmniej 3 konta o różnym zakresie dostępu
- Co najmniej 2 implementacje (MS SQL, Sybase Central)

2. Tematyka oraz założenia projektu

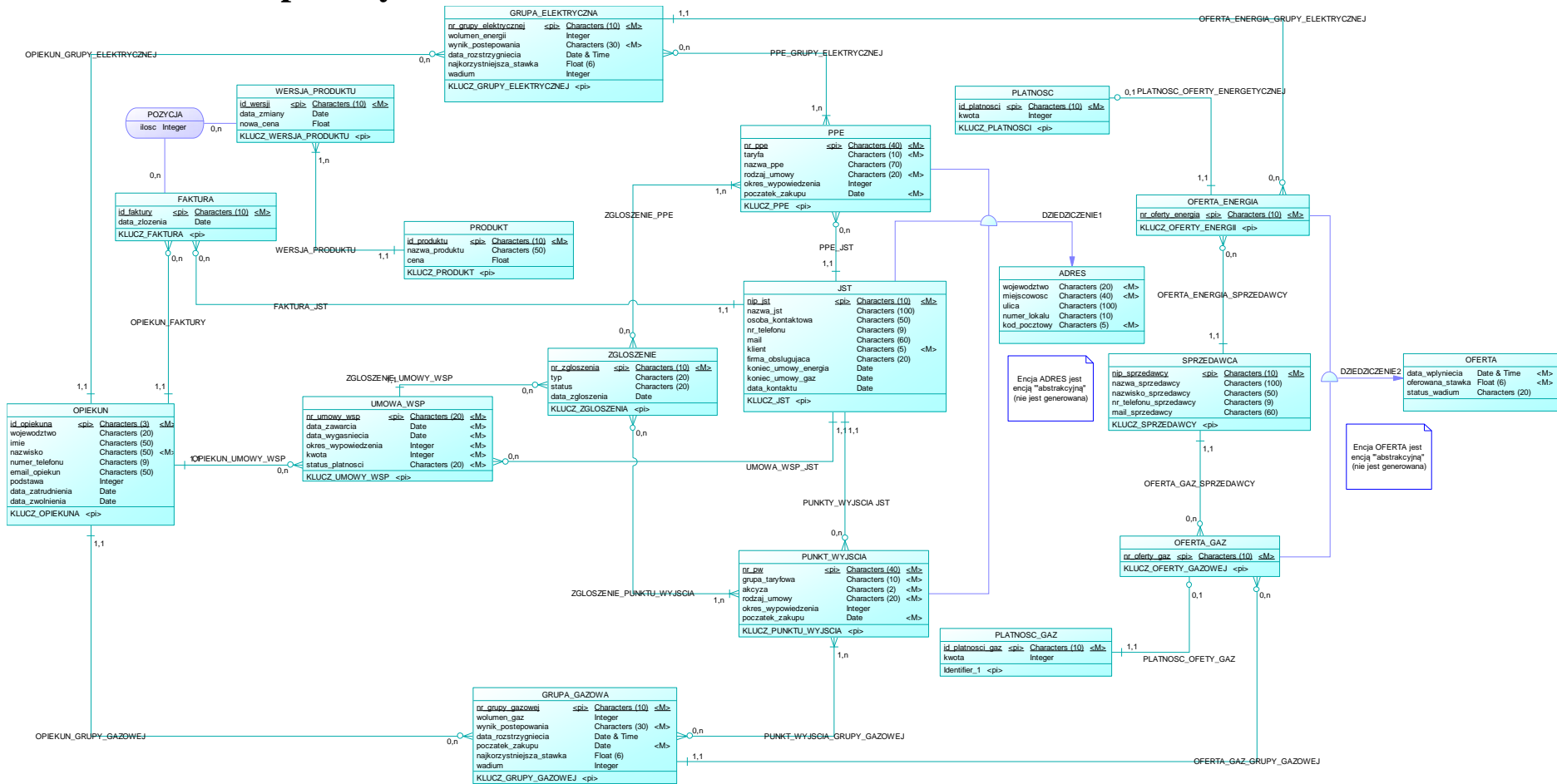
Tematem powyższego projektu jest utworzenie bazy danych realnie istniejącej firmy Voltra Energy, która zajmuje się grupowym zakupem energii elektrycznej oraz paliwa gazowego dla Jednostek Samorządowych, a także obsługą formalno-prawną po przeprowadzonym postępowaniu przetargowym. Dodatkowo firma sprzedaje oprawy oświetlenia ulicznego.

Kraj podzielony jest pośród szesnastu opiekunów klienta – każdy opiekun sprawuje pieczę nad jednym województwem. Pozyskuje nowych klientów, przygotowuje dokumentacje do przetargu, a także obsługuje zgłoszenia dodania lub usunięcia punktów poboru energii elektrycznej, punktów wyjścia do Sprzedawców w imieniu obecnych klientów.

Firma Voltra Energy swoją ofertę kieruje tylko i wyłącznie do Jednostek Samorządów Terytorialnych, w związku tym tworzone grupy zakupowe opierają się na unijnych postępowaniach przetargowych. Grupowy zakup niesie za sobą wiele korzyści dla klientów: gwarancja stałej ceny zakupu do końca trwania umowy, przygotowanie wszelkiej niezbędnej dokumentacji przez ekspertów, a także uzyskanie niższej stawki przez duże wolumeny grup.

Baza danych umożliwia bardzo szybkie i łatwe odnalezienie wszystkich informacji na temat klienta, grupy zakupowej, a także obliczenie wykonanego obrotu firmy, wyłonienie najlepszego zespołu wraz obliczonym obrotem rocznym. Dodatkowo baza automatyzuje wybór najkorzystniejszej oferty spośród wszystkich ofert złożonych przez oferentów, a także oferuje automatyczne obniżenie ceny produktów w przypadku niskich przychodów za konkretny produkt. Co więcej, silnik bazy danych pilnuje aby wprowadzane dane dotyczące punktów poboru energii elektrycznej były prawidłowe np. prognozowane zużycie nie może być mniejsze od zera, a taryfa punktu musi być taka sama jak jeden z elementów zbioru istniejących taryf. Informacje zawarte w bazie danych umożliwiają wyliczenie podwyżki na podstawie osiągniętych wyników sprzedażowych, a także odprawy w przypadku zakończenia stosunku pracy.

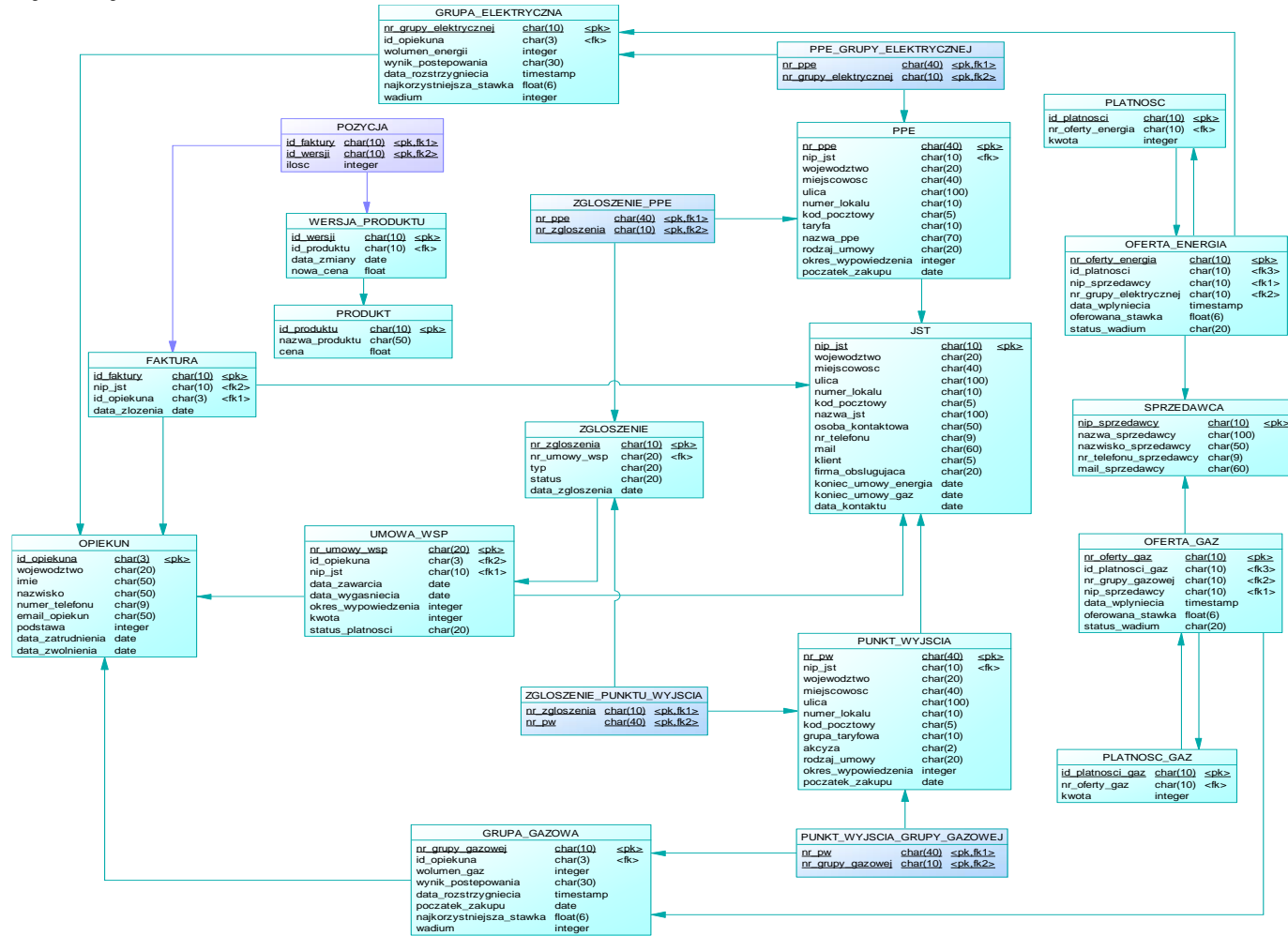
3. Model konceptualny



Model konceptualny utworzony na bazie założeń zawiera 18 encji oraz jedną asocjację. Dwie spośród encji (ADRES oraz OFERTA) są encjami abstrakcyjnymi. Model zawiera również jedno potrójne i jedno podwójne dziedziczenie.

- Encja OPIEKUN zawiera informacje na temat danych osobowych pracownika, a także nad którym województwem spracuje opiekę oraz ile zarabia. Encja powiązana jest z encjami: UMOWA_WSP, FAKTURA, GRUPA_ELEKTRYCZNA, GRUPA_GAZOWA.
- Encja JST zawiera informacje na temat obecnych oraz byłych klientów firmy, a także Jednostek Samorządów Terytorialnych z którymi kontaktowali się opiekunowie. Encja powiązana jest z encjami: PUNKT_WYJSCIA, PPE, UMOWA_WSP oraz FAKTURA.
- Encje GRUPA_ELEKTRYCZNA oraz GRUPA_GAZOWA zawierają informacje na temat prowadzonych oraz zamkniętych postępowań przetargowych wraz z wolumenami, wysokością wadium oraz najkorzystniejszą stawką. Encje powiązane są z encjami: OPIEKUN oraz odpowiednio OFERTA_ENERGIA, OFERTA_GAZ.
- Encje PPE oraz PUNKT_WYJSCIA zawierają informacje na temat odpowiednio punktów poboru energii elektrycznej oraz punktów wyjścia gazu wraz z adresem, taryfą oraz numerem identyfikacyjnym nadawanym przed dystrybucją. Encje powiązane są z encjami: JST, ZGLOSZENIE oraz odpowiednio GRUPA_ELEKTRYCZNA i PUNKT_WYJSCIA.
- Encje OFERTA_ENERGIA oraz OFERTA_GAZ zawierają informacje na temat złożonych ofert, które są odpowiedzią na prowadzone postępowanie przetargowe. Encje powiązane są z encjami: SPRZEDAWCA oraz odpowiednio GRUPA_ELEKTRYCZNA i GRUPA_GAZOWA.
- Encja PRODUKT zawiera informacje na temat ceny oraz nazwy produktu, w przypadku firmy VOLTRA są to tylko i wyłącznie oprawy oświetlenia ulicznego. Encja powiązana jest z encją WERSJA_PRODUKTU.
- Encja WERSJA_PRODUKTU przechowuje informacje na temat historii zmian cen produktów wraz z datą zmiany oraz nową ceną. Encja powiązana jest z encją PRODUKT oraz asocjacją POZYCJA.
- Encja ZGLOSZENIE zawiera informacje na temat zgłoszeń dokonywanych przez klientów w celu zaaneksowania umowy aby dołączyć lub usunąć punkt poboru energii elektrycznej lub punkt wyjścia. Encja powiązana jest z encjami: PPE, UMOWA_WSP, PUNKT_WYJSCIA.

4. Model fizyczny



5. Listing kodu generującego bazę danych

```
create table FAKTURA
(
    id_faktury          char(10)          not null,
    nip_jst             char(10)          not null,
    id_opiekuna         char(3)           not null,
    data_zlozenia       date              null,
    constraint PK_FAKTURA primary key (id_faktury)
);

comment on column FAKTURA.id_faktury is
'Jest to unikatowy numer faktury. ';

comment on column FAKTURA.nip_jst is
'Jest to dziesięciocyfrowy kod, służący do identyfikacji podatników w
Polsce.';

comment on column FAKTURA.id_opiekuna is
'Identyfikator opiekuna to unikatowy ciąg liczbowo-znakowy za pomocą,
którego można rozpoznać opiekuna.';

comment on column FAKTURA.data_zlozenia is
'Jest to data zlozenia zamowienia, a tym samym otwarcia zamowienia. ';

/*=====*/
/* Index: FAKTURA_PK                                     */
/*=====*/
create unique index FAKTURA_PK on FAKTURA (
id_faktury ASC
);

/*=====*/
/* Index: OPIEKUN_FAKTURY_FK                             */
/*=====*/
create index OPIEKUN_FAKTURY_FK on FAKTURA (
id_opiekuna ASC
);

/*=====*/
/* Index: FAKTURA_JST_FK                                 */
/*=====*/
create index FAKTURA_JST_FK on FAKTURA (
nip_jst ASC
);

/*=====*/
/* Table: GRUPA_ELEKTRYCZNA                             */
/*=====*/
create table GRUPA_ELEKTRYCZNA
(
    nr_grupy_elektrycznej char(10)          not null,
    id_opiekuna           char(3)           not null,
    wolumen_energii       integer           null,
```

```

    wynik_postepowania    char(30)                                not null default
    'Nierozstrzygniety'
    constraint CKC_WYNIK_POSTEPOWANI_GRUPA_EL check (wynik_postepowania
in ('Nierozstrzygniety', 'Rozstrzygniety', 'Brak ofert')),
    data_rozstrzygnięcia timestamp                                null,
    najkorzystniejsza_stawka float(6)                            null,
    wadium                 integer                                null,
    constraint PK_GRUPA_ELEKTRYCZNA primary key (nr_grupy_elektrycznej)
);

comment on table GRUPA_ELEKTRYCZNA is
'Grupa elektryczna jest to postępowanie przetargowe w którym bierze udział
zbiór punktów poboru energii elektrycznej jednostek samorządów
terytorialnych, które przystępują wspólnie do postępowania przetargowego
w celu osiągnięcia korzystniejszej oferty. ';

comment on column GRUPA_ELEKTRYCZNA.nr_grupy_elektrycznej is
'Jest to unikatowy numer nadawany przez firmę. ';

comment on column GRUPA_ELEKTRYCZNA.id_opiekuna is
'Identyfikator opiekuna to unikatowy ciąg liczbowo-znakowy za pomocą,
którego można rozpoznać opiekuna.';

comment on column GRUPA_ELEKTRYCZNA.wolumen_energii is
'dupa';

comment on column GRUPA_ELEKTRYCZNA.wynik_postepowania is
'Jest to wynik przeprowadzonego postępowania przetargowego.';

comment on column GRUPA_ELEKTRYCZNA.data_rozstrzygnięcia is
'Jest to data i czas zakończenia przyjmowania ofert od oferentów. ';

comment on column GRUPA_ELEKTRYCZNA.najkorzystniejsza_stawka is
'Stawka w złotych za jedną kWh najkorzystniejszej oferty. ';

comment on column GRUPA_ELEKTRYCZNA.wadium is
'Wadium to określona suma pieniędzy składana przez oferenta w poczet
zabezpieczenia dotrzymania warunków danej umowy w ramach przetargu.';

/*=====*/
/* Index: GRUPA_ELEKTRYCZNA_PK */
/*=====*/
create unique index GRUPA_ELEKTRYCZNA_PK on GRUPA_ELEKTRYCZNA (
nr_grupy_elektrycznej ASC
);

/*=====*/
/* Index: OPIEKUN_GRUPY_ELEKTRYCZNEJ_FK */
/*=====*/
create index OPIEKUN_GRUPY_ELEKTRYCZNEJ_FK on GRUPA_ELEKTRYCZNA (
id_opiekuna ASC
);

/*=====*/

```



```

/* Table: GRUPA_GAZOWA */
/*=====*/
create table GRUPA_GAZOWA
(
    nr_grupy_gazowej      char(10)          not null,
    id_opiekuna           char(3)           not null,
    wolumen_gaz           integer           null,
    wynik_postepowania    char(30)         not null default
    'Nierozstrzygniety'
    constraint CKC_WYNIK_POSTEPOWANI_GRUPA_GA check (wynik_postepowania
in ('Nierozstrzygniety', 'Rozstrzygniety', 'Brak ofert')),
    data_rozstrzygnięcia timestamp          null,
    poczatek_zakupu       date              not null,
    najkorzystniejsza_stawka float(6)       null,
    wadium               integer           null,
    constraint PK_GRUPA_GAZOWA primary key (nr_grupy_gazowej)
);

comment on table GRUPA_GAZOWA is
'Grupa gazowa jest to postępowanie przetargowe w którym bierze udział
zbiór punktów wyjścia gazu jednostek samorządów terytorialnych, które
przystępują wspólnie do postępowania przetargowego w celu osiągnięcia
korzystniejszej oferty. ';

comment on column GRUPA_GAZOWA.nr_grupy_gazowej is
'Jest to unikatowy numer identyfikujący grupę gazową.';

comment on column GRUPA_GAZOWA.id_opiekuna is
'Identyfikator opiekuna to unikatowy ciąg liczbowo-znakowy za pomocą,
którego można rozpoznać opiekuna.';

comment on column GRUPA_GAZOWA.wolumen_gaz is
'Jest to łączny prognozowany pobór paliwa gazowego podczas obowiązywania
umowy ze sprzedawcą wyrażony w kWh.';

comment on column GRUPA_GAZOWA.wynik_postepowania is
'Jest to wynik przeprowadzonego postępowania przetargowego.';

comment on column GRUPA_GAZOWA.data_rozstrzygnięcia is
'Jest to data i czas zakończenia przyjmowania ofert od oferentów. ';

comment on column GRUPA_GAZOWA.poczatek_zakupu is
'Jest to punkt wskazujący początek zakupu.';

comment on column GRUPA_GAZOWA.najkorzystniejsza_stawka is
'Stawka w złotych za jedną kWh najkorzystniejszej oferty. ';

comment on column GRUPA_GAZOWA.wadium is
'Wadium to określona suma pieniędzy składana przez oferenta w poczet
zabezpieczenia dotrzymania warunków danej umowy w ramach przetargu.';

/*=====*/
/* Index: GRUPA_GAZOWA_PK */
/*=====*/

```

```

create unique index GRUPA_GAZOWA_PK on GRUPA_GAZOWA (
nr_grupy_gazowej ASC
);

/*=====*/
/* Index: OPIEKUN_GRUPY_GAZOWEJ_FK */
/*=====*/
create index OPIEKUN_GRUPY_GAZOWEJ_FK on GRUPA_GAZOWA (
id_opiekuna ASC
);

/*=====*/
/* Table: JST */
/*=====*/
create table JST
(
    nip_jst                char(10)                not null,
    wojewodztwo            char(20)                not null,
    miejscowosc            char(40)                not null,
    ulica                  char(100)               null,
    numer_lokalu           char(10)                null,
    kod_pocztowy           char(5)                 not null,
    nazwa_jst              char(100)              null,
    osoba_kontaktowa      char(50)                null,
    nr_telefonu            char(9)                 null,
    mail                   char(60)                null,
    klient                 char(5)                 not null
        constraint CKC_KLIENT_JST check (klient in ('Tak','Nie')),
    firma_obsługujaca      char(20)                null,
    koniec_umowy_energia   date                    null,
    koniec_umowy_gaz       date                    null,
    data_kontakt          date                     null,
    constraint PK_JST primary key (nip_jst)
);

comment on table JST is
'JST jest to Jednostka Samorządu Terytorialnego czyli organizacja
społeczności lokalnej lub regionalnej i jednocześnie forma administracji
publicznej,';

comment on column JST.nip_jst is
'Jest to dziesięciocyfrowy kod, służący do identyfikacji podatników w
Polsce.';

comment on column JST.wojewodztwo is
'Województwo jest to jednostka terytorialna, której opiekunem jest dany
pracownik (opiekun).';

comment on column JST.miejscowosc is
'Jest to nazwa miejscowości w której znajduje się punkt.';

comment on column JST.ulica is
'Jest to nazwa ulicy na której znajduje się punkt.';

```

```

comment on column JST.numer_lokalu is
'Jest to numer lokalu pod którym znajduje się punkt.';

comment on column JST.kod_pocztowy is
'Jest to kod pocztowy miejscowosci w której znajduje się punkt.';

comment on column JST.nazwa_jst is
'Jest to nazwa jednostki samorządu terytorialnego. ';

comment on column JST.osoba_kontaktowa is
'Jest to nazwisko osoby, która zajmuje się od strony jednostki samorządu
terytorialnego zagadnieniami związanymi z przeprowadzaniem przetargów. ';

comment on column JST.nr_telefonu is
'Jest to numer telefonu do osoby kontaktowej w jednostrze samorządu
terytorialnego. ';

comment on column JST.mail is
'Jest to adres mail do osoby kontaktowej w jednostrze samorządu
terytorialnego. ';

comment on column JST.klient is
'Jest to określenie czy dana jednostka samorządu terytorialnego jest
klientem firmy. ';

comment on column JST.firma_obslugujaca is
'Jest to nazwa firmy, która zajmuje się obsługą formalno-prawną energii
elektrycznej i gazu dla jednostki samorządu terytorialnego. ';

comment on column JST.koniec_umowy_energia is
'Jest to data zakończenia umowy sprzedaży energii elektrycznej dla
jednostki samorządu terytorialnego. ';

comment on column JST.koniec_umowy_gaz is
'Jest to data zakończenia umowy sprzedaży paliwa gazowego dla jednostki
samorządu terytorialnego. ';

comment on column JST.data_kontaktu is
'Jest to data ostatniego kontaktu przez opiekuna województwa z osobą
kontaktową jednostki samorządu terytorialnego. ';

/*=====*/
/* Index: JST_PK */
/*=====*/
create unique index JST_PK on JST (
nip_jst ASC
);

/*=====*/
/* Table: OFERTA_ENERGIA */
/*=====*/
create table OFERTA_ENERGIA
(
    nr_oferty_energia    char(10)                not null,

```

```

id_platnosci          char(10)          null,
nip_sprzedawcy        char(10)          not null,
nr_grupy_elektrycznej char(10)          not null,
data_wplyniecia       timestamp         not null,
oferowana_stawka      float(6)          not null,
status_wadium         char(20)          null,
constraint PK_OFERTA_ENERGIA primary key (nr_oferty_energia)
);

comment on table OFERTA_ENERGIA is
'Jesto to oferta skłanada w odpowiedzi na przetarg przez sprzedawcę.';

comment on column OFERTA_ENERGIA.nr_oferty_energia is
'Jest to indywidulany numer oferty nadawany po wpłynięciu oferty.';

comment on column OFERTA_ENERGIA.id_platnosci is
'Jest to unikatowy ciąg znaków charakteryzujący każdą płatność wpływającą
do firmy';

comment on column OFERTA_ENERGIA.nip_sprzedawcy is
'Jest to numer identyfikacji podatkowej sprzedawcy. ';

comment on column OFERTA_ENERGIA.nr_grupy_elektrycznej is
'Jest to unikatowy numer nadawany przez firmę. ';

comment on column OFERTA_ENERGIA.data_wplyniecia is
'Jest to dokładna data i czas wpłynięcia oferty. ';

comment on column OFERTA_ENERGIA.oferowana_stawka is
'Jest to stawka netto oferowana przez sprzedawcę za kWh. ';

comment on column OFERTA_ENERGIA.status_wadium is
'Jest to status wadium wpłacanego w poczet zabezpieczenia składanej oferty
przez oferentów. ';

/*=====*/
/* Index: OFERTA_ENERGIA_PK */
/*=====*/
create unique index OFERTA_ENERGIA_PK on OFERTA_ENERGIA (
nr_oferty_energia ASC
);

/*=====*/
/* Index: OFERTA_ENERGIA_SPRZEDAWCY_FK */
/*=====*/
create index OFERTA_ENERGIA_SPRZEDAWCY_FK on OFERTA_ENERGIA (
nip_sprzedawcy ASC
);

/*=====*/
/* Index: OFERTA_ENERGIA_GRUPY_ELEKTRYCZNEJ_FK */
/*=====*/
create index OFERTA_ENERGIA_GRUPY_ELEKTRYCZNEJ_FK on OFERTA_ENERGIA (
nr_grupy_elektrycznej ASC

```

```

);

/*=====*/
/* Index: PLATNOSC_OFERTY_ENERGETYCZNEJ2_FK */
/*=====*/
create index PLATNOSC_OFERTY_ENERGETYCZNEJ2_FK on OFERTA_ENERGIA (
id_platnosci ASC
);

/*=====*/
/* Table: OFERTA_GAZ */
/*=====*/
create table OFERTA_GAZ
(
    nr_oferty_gaz          char(10)          not null,
    id_platnosci_gaz       char(10)          not null,
    nr_grupy_gazowej       char(10)          not null,
    nip_sprzedawcy         char(10)          not null,
    data_wplyniecia        timestamp         not null,
    oferowana_stawka       float(6)          not null,
    status_wadium          char(20)          null,
    constraint PK_OFERTA_GAZ primary key (nr_oferty_gaz)
);

comment on table OFERTA_GAZ is
'Jesto to oferta skłanada w odpowiedzi na przetarg przez sprzedawcę.';

comment on column OFERTA_GAZ.nr_oferty_gaz is
'Jest to unikatowy numer identyfikujący ofertę składaną przez oferenta. ';

comment on column OFERTA_GAZ.id_platnosci_gaz is
'Jest to unikatowy ciąg znaków charakteryzujący każdą płatność wpływającą
do firmy';

comment on column OFERTA_GAZ.nr_grupy_gazowej is
'Jest to unikatowy numer identyfikujący grupę gazową.';

comment on column OFERTA_GAZ.nip_sprzedawcy is
'Jest to numer identyfikacji podatkowej sprzedawcy. ';

comment on column OFERTA_GAZ.data_wplyniecia is
'Jest to dokładna data i czas wplynięcia oferty. ';

comment on column OFERTA_GAZ.oferowana_stawka is
'Jest to stawka netto oferowana przez sprzedawcę za kWh. ';

comment on column OFERTA_GAZ.status_wadium is
'Jest to status wadium wplacanego w poczet zabezpieczenia składanej ofety
przez oferentów. ';

/*=====*/
/* Index: OFERTA_GAZ_PK */
/*=====*/
create unique index OFERTA_GAZ_PK on OFERTA_GAZ (

```

```

nr_oferty_gaz ASC
);

/*=====*/
/* Index: OFERTA_GAZ_SPRZEDAWCY_FK */
/*=====*/
create index OFERTA_GAZ_SPRZEDAWCY_FK on OFERTA_GAZ (
nip_sprzedawcy ASC
);

/*=====*/
/* Index: OFERTA_GAZ_GRUPY_GAZOWEJ_FK */
/*=====*/
create index OFERTA_GAZ_GRUPY_GAZOWEJ_FK on OFERTA_GAZ (
nr_grupy_gazowej ASC
);

/*=====*/
/* Index: PLATNOSC_OFETY_GAZ2_FK */
/*=====*/
create index PLATNOSC_OFETY_GAZ2_FK on OFERTA_GAZ (
id_platnosci_gaz ASC
);

/*=====*/
/* Table: OPIEKUN */
/*=====*/
create table OPIEKUN
(
    id_opiekuna          char(3)          not null,
    wojewodztwo          char(20)         null,
    imie                 char(50)         null,
    nazwisko             char(50)         not null,
    numer_telefonu        char(9)         null,
    email_opiekun         char(50)        null,
    podstawa             integer          null,
    data_zatrudnienia     date            null,
    data_zwolnienia       date            null,
    constraint PK_OPIEKUN primary key (id_opiekuna)
);

comment on table OPIEKUN is
'Opiekun jest to pracownik działu handlowego zajmujący się pozyskiwaniem
nowych klientów oraz obsługą formalno-prawną obecnych klientów. ';

comment on column OPIEKUN.id_opiekuna is
'Identyfikator opiekuna to unikatowy ciąg liczbowo-znakowy za pomocą,
którego można rozpoznać opiekuna.';

comment on column OPIEKUN.wojewodztwo is
'Województwo jest to jednostka terytorialna, której opiekunem jest dany
pracownik (opiekun).';

comment on column OPIEKUN.imie is

```

```

'Imie opiekuna od strony firmy.';

comment on column OPIEKUN.nazwisko is
'Nazwisko opiekuna od strony firmy.';

comment on column OPIEKUN.numer_telefonu is
'Numer służbowy opiekuna od strony firmy.';

comment on column OPIEKUN.email_opiekun is
'Email jest to system przesyłania informacji w sieci komputerowej.';

comment on column OPIEKUN.podstawa is
'Jest to wynagrodzenie wypłacane w każdym miesiącu za przepracowany
poprzedni miesiąc.';

comment on column OPIEKUN.data_zatrudnienia is
'Jest to data nawiązania stosunku pracy pomiędzy pracownikiem, a
pracodawcą.';

comment on column OPIEKUN.data_zwolnienia is
'Jest to data zaprzestania stosunku pracy pomiędzy pracownikiem, a
pracodawcą.';

/*=====*/
/* Index: OPIEKUN_PK */
/*=====*/
create unique index OPIEKUN_PK on OPIEKUN (
id_opiekuna ASC
);

/*=====*/
/* Table: PLATNOSC */
/*=====*/
create table PLATNOSC
(
    id_platnosci      char(10)          not null,
    nr_oferty_energia char(10)          not null,
    kwota             integer           null,
    constraint PK_PLATNOSC primary key (id_platnosci)
);

comment on table PLATNOSC is
'Płatność jest to obowiązek regulowania zobowiązań pieniężnych w
określonym terminie';

comment on column PLATNOSC.id_platnosci is
'Jest to unikatowy ciąg znaków charakteryzujący każdą płatność wpływającą
do firmy';

comment on column PLATNOSC.nr_oferty_energia is
'Jest to indywidualny numer oferty nadawany po wpłynięciu oferty.';

comment on column PLATNOSC.kwota is

```

```

'Jest to wartość netto jaką JST zobowiązało się wypłacić na poczet firmy
za wykonane usługi. ';

/*=====*/
/* Index: PLATNOSC_PK */
/*=====*/
create unique index PLATNOSC_PK on PLATNOSC (
id_platnosci ASC
);

/*=====*/
/* Index: PLATNOSC_OFERTY_ENERGETYCZNEJ_FK */
/*=====*/
create index PLATNOSC_OFERTY_ENERGETYCZNEJ_FK on PLATNOSC (
nr_oferty_energia ASC
);

/*=====*/
/* Table: PLATNOSC_GAZ */
/*=====*/
create table PLATNOSC_GAZ
(
    id_platnosci_gaz      char(10)                not null,
    nr_oferty_gaz         char(10)                null,
    kwota                 integer                 null,
    constraint PK_PLATNOSC_GAZ primary key (id_platnosci_gaz)
);

comment on column PLATNOSC_GAZ.id_platnosci_gaz is
'Jest to unikatowy ciąg znaków charakteryzujący każdą płatność wpływającą
do firmy';

comment on column PLATNOSC_GAZ.nr_oferty_gaz is
'Jest to unikatowy numer identyfikujący ofertę składaną przez oferenta. ';

comment on column PLATNOSC_GAZ.kwota is
'Jest to wartość netto jaką JST zobowiązało się wypłacić na poczet firmy
za wykonane usługi. ';

/*=====*/
/* Index: PLATNOSC_GAZ_PK */
/*=====*/
create unique index PLATNOSC_GAZ_PK on PLATNOSC_GAZ (
id_platnosci_gaz ASC
);

/*=====*/
/* Index: PLATNOSC_OFETY_GAZ_FK */
/*=====*/
create index PLATNOSC_OFETY_GAZ_FK on PLATNOSC_GAZ (
nr_oferty_gaz ASC
);

/*=====*/

```



```

/* Table: POZYCJA */
/*=====*/
create table POZYCJA
(
    id_faktury          char(10)          not null,
    id_wersji           char(10)          not null,
    ilosc               integer            null,
    constraint PK_POZYCJA primary key (id_faktury, id_wersji)
);

comment on column POZYCJA.id_faktury is
'Jest to unikatowy numer faktury. ';

comment on column POZYCJA.id_wersji is
'Jest to unikatowy numer wersji produktu. ';

comment on column POZYCJA.ilosc is
'Jest to ilość produktów ';

/*=====*/
/* Index: POZYCJA_PK */
/*=====*/
create unique index POZYCJA_PK on POZYCJA (
id_faktury ASC,
id_wersji ASC
);

/*=====*/
/* Index: POZYCJA_FK */
/*=====*/
create index POZYCJA_FK on POZYCJA (
id_faktury ASC
);

/*=====*/
/* Index: POZYCJA2_FK */
/*=====*/
create index POZYCJA2_FK on POZYCJA (
id_wersji ASC
);

/*=====*/
/* Table: PPE */
/*=====*/
create table PPE
(
    nr_ppe              char(40)          not null,
    nip_jst             char(10)          not null,
    wojewodztwo         char(20)          not null,
    miejscowosc         char(40)          not null,
    ulica               char(100)         null,
    numer_lokalu        char(10)          null,
    kod_pocztowy        char(5)          not null,
    taryfa              char(10)          not null,

```

```

nazwa_ppe          char(70)          null,
rodzaj_umowy       char(20)          not null
    constraint CKC_RODZAJ_UMOWY_PPE check (rodzaj_umowy in
('Kompleksowa','Rozdzielna')),
okres_wypowiedzenia integer          null,
poczatek_zakupu    date              not null,
    constraint PK_PPE primary key (nr_ppe)
);

comment on table PPE is
'Jest to punkt poboru energii elektrycznej czyli najmniejsza jednostka,
dla której odbywa się zbilansowanie dostaw energii elektrycznej, oraz dla
której może nastąpić zmiana sprzedawcy.';

comment on column PPE.nr_ppe is
'Jest to unikatowy ciąg liczbowo-znakowy ppe, pozwalający go
scharakteryzować. ';

comment on column PPE.nip_jst is
'Jest to dziesięciocyfrowy kod, służący do identyfikacji podatników w
Polsce.';

comment on column PPE.wojewodztwo is
'Województwo jest to jednostka terytorialna, której opiekunem jest dany
pracownik (opiekun).';

comment on column PPE.miejscowosc is
'Jest to nazwa miejscowości w której znajduje się punkt.';

comment on column PPE.ulica is
'Jest to nazwa ulicy na której znajduje się punkt.';

comment on column PPE.numer_lokalu is
'Jest to numer lokalu pod którym znajduje się punkt.';

comment on column PPE.kod_pocztowy is
'Jest to kod pocztowy miejscowosci w której znajduje się punkt.';

comment on column PPE.taryfa is
'Jest to plan cenowy zgodnie z którym odbiorca płaci. ';

comment on column PPE.nazwa_ppe is
'Jest to nazwa obiektu, którego dotyczy punkt. ';

comment on column PPE.rodzaj_umowy is
'Jest to rodzaj umowy jaka jest zawarta obecnie na dany punkt. ';

comment on column PPE.okres_wypowiedzenia is
'Okres wypowiedzenia umowy współpracy między firmą, a JST wyrażony w
miesiącach.';

comment on column PPE.poczatek_zakupu is
'Jest to punkt wskazujący początek zakupu.';

```

```

/*=====*/
/* Index: PPE_PK */
/*=====*/
create unique index PPE_PK on PPE (
nr_ppe ASC
);

/*=====*/
/* Index: PPE_JST_FK */
/*=====*/
create index PPE_JST_FK on PPE (
nip_jst ASC
);

/*=====*/
/* Table: PPE_GRUPY_ELEKTRYCZNEJ */
/*=====*/
create table PPE_GRUPY_ELEKTRYCZNEJ
(
    nr_ppe char(40) not null,
    nr_grupy_elektrycznej char(10) not null,
    constraint PK_PPE_GRUPY_ELEKTRYCZNEJ primary key (nr_ppe,
nr_grupy_elektrycznej)
);

comment on table PPE_GRUPY_ELEKTRYCZNEJ is
'Jest to związek łączący encję PPE i GRUPA_ELEKTRYCZNA. Grupa elektryczna
może posiadać wiele PPE, a PPE może znajdować się w wielu grupach
elektrycznych.';

comment on column PPE_GRUPY_ELEKTRYCZNEJ.nr_ppe is
'Jest to unikatowy ciąg liczbowo-znakowy ppe, pozwalający go
scharakteryzować. ';

comment on column PPE_GRUPY_ELEKTRYCZNEJ.nr_grupy_elektrycznej is
'Jest to unikatowy numer nadawany przez firmę. ';

/*=====*/
/* Index: PPE_GRUPY_ELEKTRYCZNEJ_PK */
/*=====*/
create unique index PPE_GRUPY_ELEKTRYCZNEJ_PK on PPE_GRUPY_ELEKTRYCZNEJ (
nr_ppe ASC,
nr_grupy_elektrycznej ASC
);

/*=====*/
/* Index: PPE_GRUPY_ELEKTRYCZNEJ_FK */
/*=====*/
create index PPE_GRUPY_ELEKTRYCZNEJ_FK on PPE_GRUPY_ELEKTRYCZNEJ (
nr_ppe ASC
);

/*=====*/
/* Index: PPE_GRUPY_ELEKTRYCZNEJ2_FK */
/*=====*/

```

```

/*=====*/
create index PPE_GRUPY_ELEKTRYCZNEJ2_FK on PPE_GRUPY_ELEKTRYCZNEJ (
nr_grupy_elektrycznej ASC
);

/*=====*/
/* Table: PRODUKT */
/*=====*/
create table PRODUKT
(
    id_produktu          char(10)          not null,
    nazwa_produktu       char(50)          null,
    cena                 float             null,
    constraint PK_PRODUKT primary key (id_produktu)
);

comment on table PRODUKT is
'Produkt czyli to, co zostało wyprodukowane i jest przeznaczone na
sprzedaż';

comment on column PRODUKT.id_produktu is
'Jest to unikatowy numer identyfikujący produkt.';

comment on column PRODUKT.nazwa_produktu is
'Jest to nazwa produktu. ';

comment on column PRODUKT.cena is
'Jesto to cena netto za produkt, ';

/*=====*/
/* Index: PRODUKT_PK */
/*=====*/
create unique index PRODUKT_PK on PRODUKT (
id_produktu ASC
);

/*=====*/
/* Table: PUNKT_WYJSCIA */
/*=====*/
create table PUNKT_WYJSCIA
(
    nr_pw                char(40)          not null,
    nip_jst              char(10)          not null,
    wojewodztwo          char(20)          not null,
    miejscowosc          char(40)          not null,
    ulica                char(100)         null,
    numer_lokalu          char(10)          null,
    kod_pocztowy          char(5)           not null,
    grupa_taryfowa        char(10)          not null,
    akcyza                char(2)           not null
    constraint CKC_AKCYZA_PUNKT_WY check (akcyza in
('Zwolniony','Płaci')),
    rodzaj_umowy          char(20)         not null
);

```

```

        constraint CKC_RODZAJ_UMOWY_PUNKT_WY check (rodzaj_umowy in
('Kompleksowa','Rozdzielna')),
        okres_wypowiedzenia integer null,
        poczatek_zakupu date not null,
        constraint PK_PUNKT_WYJSCIA primary key (nr_pw)
);

comment on table PUNKT_WYJSCIA is
'Jest to najmniejsza jednostka, dla której odbywa się zbilansowanie dostaw
paliwa gazowego, oraz dla której może nastąpić zmiana sprzedawcy.';

comment on column PUNKT_WYJSCIA.nr_pw is
'Jest to unikatowy ciąg liczbowo-znakowy punktu wyjścia, pozwalający go
scharakteryzować. ';

comment on column PUNKT_WYJSCIA.nip_jst is
'Jest to dziesięciocyfrowy kod, służący do identyfikacji podatników w
Polsce.';

comment on column PUNKT_WYJSCIA.wojewodztwo is
'Województwo jest to jednostka terytorialna, której opiekunem jest dany
pracownik (opiekun).';

comment on column PUNKT_WYJSCIA.miejscowosc is
'Jest to nazwa miejscowości w której znajduje się punkt.';

comment on column PUNKT_WYJSCIA.ulica is
'Jest to nazwa ulicy na której znajduje się punkt.';

comment on column PUNKT_WYJSCIA.numer_lokalu is
'Jest to numer lokalu pod którym znajduje się punkt.';

comment on column PUNKT_WYJSCIA.kod_pocztowy is
'Jest to kod pocztowy miejscowosci w której znajduje się punkt.';

comment on column PUNKT_WYJSCIA.grupa_taryfowa is
'Jest to plan cenowy zgodnie z którym odbiorca płaci za paliwo gazowe.';

comment on column PUNKT_WYJSCIA.akcyza is
'Jest to określenie czy dany punkt jest zwolniony z płacenia akcyzy czy
jednak akcyza musi być płacona za dany punkt. ';

comment on column PUNKT_WYJSCIA.rodzaj_umowy is
'Jest to rodzaj umowy jaka jest zawarta obecnie na dany punkt. ';

comment on column PUNKT_WYJSCIA.okres_wypowiedzenia is
'Okres wypowiedzenia umowy współpracy między firmą, a JST wyrażony w
miesiącach.';

comment on column PUNKT_WYJSCIA.poczatek_zakupu is
'Jest to punkt wskazujący początek zakupu.';

/*=====*/
/* Index: PUNKT_WYJSCIA_PK */

```

```

/*=====*/
create unique index PUNKT_WYJSCIA_PK on PUNKT_WYJSCIA (
nr_pw ASC
);

/*=====*/
/* Index: "PUNKTY_WYJSCIA_JST_FK" */
/*=====*/
create index "PUNKTY_WYJSCIA_JST_FK" on PUNKT_WYJSCIA (
nip_jst ASC
);

/*=====*/
/* Table: PUNKT_WYJSCIA_GRUPY_GAZOWEJ */
/*=====*/
create table PUNKT_WYJSCIA_GRUPY_GAZOWEJ
(
    nr_pw                char(40)                not null,
    nr_grupy_gazowej     char(10)                not null,
    constraint PK_PUNKT_WYJSCIA_GRUPY_GAZOWEJ primary key (nr_pw,
nr_grupy_gazowej)
);

comment on table PUNKT_WYJSCIA_GRUPY_GAZOWEJ is
'Jest to związek łączący encję PUNKT_WYJSCIA i GRUPA_GAZOWA. Grupa gazowa
może posiadać wiele punktów wyjścia, a punkt wyjścia może znajdować się w
wielu grupach gazowych.';

comment on column PUNKT_WYJSCIA_GRUPY_GAZOWEJ.nr_pw is
'Jest to unikatowy ciąg liczbowo-znakowy punktu wyjścia, pozwalający go
scharakteryzować. ';

comment on column PUNKT_WYJSCIA_GRUPY_GAZOWEJ.nr_grupy_gazowej is
'Jest to unikatowy numer identyfikujący grupę gazową.';

/*=====*/
/* Index: PUNKT_WYJSCIA_GRUPY_GAZOWEJ_PK */
/*=====*/
create unique index PUNKT_WYJSCIA_GRUPY_GAZOWEJ_PK on
PUNKT_WYJSCIA_GRUPY_GAZOWEJ (
nr_pw ASC,
nr_grupy_gazowej ASC
);

/*=====*/
/* Index: PUNKT_WYJSCIA_GRUPY_GAZOWEJ_FK */
/*=====*/
create index PUNKT_WYJSCIA_GRUPY_GAZOWEJ_FK on PUNKT_WYJSCIA_GRUPY_GAZOWEJ
(
nr_pw ASC
);

/*=====*/
/* Index: PUNKT_WYJSCIA_GRUPY_GAZOWEJ2_FK */
/*=====*/

```

```

/*=====*/
create index PUNKT_WYJSCIA_GRUPY_GAZOWEJ2_FK on
PUNKT_WYJSCIA_GRUPY_GAZOWEJ (
nr_grupy_gazowej ASC
);

/*=====*/
/* Table: SPRZEDAWCA */
/*=====*/
create table SPRZEDAWCA
(
    nip_sprzedawcy          char(10)                not null,
    nazwa_sprzedawcy        char(100)               null,
    nazwisko_sprzedawcy     char(50)                null,
    nr_telefonu_sprzedawcy   char(9)                 null,
    mail_sprzedawcy          char(60)                null,
    constraint PK_SPRZEDAWCA primary key (nip_sprzedawcy)
);

comment on table SPRZEDAWCA is
'Jest to firma mająca odpowiednie pozwolenia oraz koncesje na prowadzenie
sprzedaży energii elektrycznej lub/i paliwa gazowego. ';

comment on column SPRZEDAWCA.nip_sprzedawcy is
'Jest to numer identyfikacji podatkowej sprzedawcy. ';

comment on column SPRZEDAWCA.nazwa_sprzedawcy is
'Jest to nazwa firmy świadczącej usługi sprzedaży energii elektrycznej
lub/i gazu.';

comment on column SPRZEDAWCA.nazwisko_sprzedawcy is
'Jest to nazwisko opiekuna od strony sprzedawcy. ';

comment on column SPRZEDAWCA.nr_telefonu_sprzedawcy is
'Jest to numer telefonu opiekuna od strony sprzedawcy. ';

comment on column SPRZEDAWCA.mail_sprzedawcy is
'Jest to mail opiekuna od strony sprzedawcy. ';

/*=====*/
/* Index: SPRZEDAWCA_PK */
/*=====*/
create unique index SPRZEDAWCA_PK on SPRZEDAWCA (
nip_sprzedawcy ASC
);

/*=====*/
/* Table: UMOWA_WSP */
/*=====*/
create table UMOWA_WSP
(
    nr_umowy_wsp            char(20)                not null,
    id_opiekuna              char(3)                 not null,
    nip_jst                  char(10)                not null,

```

```

data_zawarcia      date          not null,
data_wygasniecia   date          not null,
okres_wypowiedzenia integer      not null,
kwota              integer      not null,
status_platnosci   char(20)      not null,
constraint PK_UMOWA_WSP primary key (nr_umowy_wsp)
);

comment on table UMOWA_WSP is
'UMOWA_WSP jest to umowa współpracy zawierana pomiędzy firmą, a Jednostką
Samorządu Terytorialnego.';

comment on column UMOWA_WSP.nr_umowy_wsp is
'Jest to unikatowy numer umowy pomiędzy firmą, a jednostką samorządu
terytorialnego. ';

comment on column UMOWA_WSP.id_opiekuna is
'Identyfikator opiekuna to unikatowy ciąg liczbowo-znakowy za pomocą,
którego można rozpoznać opiekuna.';

comment on column UMOWA_WSP.nip_jst is
'Jest to dziesięciocyfrowy kod, służący do identyfikacji podatników w
Polsce.';

comment on column UMOWA_WSP.data_zawarcia is
'Jest to data zawarcia umowy firma-JST.';

comment on column UMOWA_WSP.data_wygasniecia is
'Jest to data wygaśnięcia umowy firma-JST.';

comment on column UMOWA_WSP.okres_wypowiedzenia is
'Okres wypowiedzenia umowy współpracy między firmą, a JST wyrażony w
miesiącach.';

comment on column UMOWA_WSP.kwota is
'Jest to wartość netto jaką JST zobowiązało się wypłacić na poczet firmy
za wykonane usługi. ';

comment on column UMOWA_WSP.status_platnosci is
'Jest to aktualny status płatności za usługi wykonane na poczet firmy
przez JST.';

/*=====*/
/* Index: UMOWA_WSP_PK */
/*=====*/
create unique index UMOWA_WSP_PK on UMOWA_WSP (
nr_umowy_wsp ASC
);

/*=====*/
/* Index: UMOWA_WSP_JST_FK */
/*=====*/
create index UMOWA_WSP_JST_FK on UMOWA_WSP (
nip_jst ASC

```



```

);

/*=====*/
/* Index: OPIEKUN_UMOWY_WSP_FK */
/*=====*/
create index OPIEKUN_UMOWY_WSP_FK on UMOWA_WSP (
id_opiekuna ASC
);

/*=====*/
/* Table: WERSJA_PRODUKTU */
/*=====*/
create table WERSJA_PRODUKTU
(
    id_wersji          char(10)          not null,
    id_produktu        char(10)          not null,
    data_zmiany        date              null,
    nowa_cena          float             null,
    constraint PK_WERSJA_PRODUKTU primary key (id_wersji)
);

comment on table WERSJA_PRODUKTU is
'Jest to tabela przechowująca historię zmian cen produktów. ';

comment on column WERSJA_PRODUKTU.id_wersji is
'Jest to unikatowy numer wersji produktu. ';

comment on column WERSJA_PRODUKTU.id_produktu is
'Jest to unikatowy numer identyfikujący produkt.';

comment on column WERSJA_PRODUKTU.data_zmiany is
'Jest to data wprowadzenia zmiany ceny produktu.';

comment on column WERSJA_PRODUKTU.nowa_cena is
'Jest to cena po zmianie. ';

/*=====*/
/* Index: WERSJA_PRODUKTU_PK */
/*=====*/
create unique index WERSJA_PRODUKTU_PK on WERSJA_PRODUKTU (
id_wersji ASC
);

/*=====*/
/* Index: WERSJA_PRODUKTU_FK */
/*=====*/
create index WERSJA_PRODUKTU_FK on WERSJA_PRODUKTU (
id_produktu ASC
);

/*=====*/
/* Table: ZGLOSZENIE */
/*=====*/
create table ZGLOSZENIE

```

```

(
    nr_zgloszenia      char(10)                not null,
    nr_umowy_wsp        char(20)                not null,
    typ                 char(20)                null
        constraint CKC_TYP_ZGLOSZEN check (typ is null or (typ in
('Dodanie','Usuniecie'))),
    status              char(20)                null,
    data_zgloszenia     date                    null,
    constraint PK_ZGLOSZENIE primary key (nr_zgloszenia)
);

comment on table ZGLOSZENIE is
'Jest to zawiadomienie o powstaniu nowego punktu poboru energii
elektrycznej/punktu wyjścia gazu lub usunięciu nowego punktu poboru energii
elektrycznej/punktu wyjścia gazu w jednostce samorządu terytorialnego. ';

comment on column ZGLOSZENIE.nr_zgloszenia is
'Jest to unikatowy ciąg liczbowo-znakowy pozwalający zidentyfikować
zgłoszenie. ';

comment on column ZGLOSZENIE.nr_umowy_wsp is
'Jest to unikatowy numer umowy pomiędzy firmą, a jednostką samorządu
terytorialnego. ';

comment on column ZGLOSZENIE.typ is
'Jest to rodzaj zgłoszenia - dodanie punktu/usunięcie punktu';

comment on column ZGLOSZENIE.status is
'Jest to obecny status trwającej procedury dołączenia/usunięcia punktu
poboru/punktu wyjścia. ';

comment on column ZGLOSZENIE.data_zgloszenia is
'Jest to data zgłoszenia dokonanego przez jednostkę samorządu
terytorialnego. ';

/*=====*/
/* Index: ZGLOSZENIE_PK                                */
/*=====*/
create unique index ZGLOSZENIE_PK on ZGLOSZENIE (
nr_zgloszenia ASC
);

/*=====*/
/* Index: ZGLOSZENIE_UMOWY_WSP_FK                        */
/*=====*/
create index ZGLOSZENIE_UMOWY_WSP_FK on ZGLOSZENIE (
nr_umowy_wsp ASC
);

/*=====*/
/* Table: ZGLOSZENIE_PPE                                */
/*=====*/
create table ZGLOSZENIE_PPE
(

```

```

    nr_ppe          char(40)          not null,
    nr_zgloszenia   char(10)          not null,
    constraint PK_ZGLOSZENIE_PPE primary key (nr_ppe, nr_zgloszenia)
);

comment on table ZGLOSZENIE_PPE is
'Jest to związek łączący encję ZGLOSZENIE i PPE. Zgłoszenie może posiadać
wiele PPE, a PPE może znajdować się w wielu zgłoszeniach.';

comment on column ZGLOSZENIE_PPE.nr_ppe is
'Jest to unikatowy ciąg liczbowo-znakowy ppe, pozwalający go
scharakteryzować. ';

comment on column ZGLOSZENIE_PPE.nr_zgloszenia is
'Jest to unikatowy ciąg liczbowo-znakowy pozwalający zidentyfikować
zgłoszenie. ';

/*=====*/
/* Index: ZGLOSZENIE_PPE_PK */
/*=====*/
create unique index ZGLOSZENIE_PPE_PK on ZGLOSZENIE_PPE (
nr_ppe ASC,
nr_zgloszenia ASC
);

/*=====*/
/* Index: ZGLOSZENIE_PPE_FK */
/*=====*/
create index ZGLOSZENIE_PPE_FK on ZGLOSZENIE_PPE (
nr_ppe ASC
);

/*=====*/
/* Index: ZGLOSZENIE_PPE2_FK */
/*=====*/
create index ZGLOSZENIE_PPE2_FK on ZGLOSZENIE_PPE (
nr_zgloszenia ASC
);

/*=====*/
/* Table: ZGLOSZENIE_PUNKTU_WYJSCIA */
/*=====*/
create table ZGLOSZENIE_PUNKTU_WYJSCIA
(
    nr_zgloszenia   char(10)          not null,
    nr_pw           char(40)          not null,
    constraint PK_ZGLOSZENIE_PUNKTU_WYJSCIA primary key (nr_zgloszenia,
nr_pw)
);

comment on table ZGLOSZENIE_PUNKTU_WYJSCIA is
'Jest to związek łączący encję ZGLOSZENIE i PUNKT_WYJSCIA. Zgłoszenie może
posiadać wiele punktów wyjścia, a punkt wyjścia może znajdować się w wielu
zgłoszeniach.';

```

```

comment on column ZGLOSZENIE_PUNKTU_WYJSCIA.nr_zgloszenia is
'Jest to unikatowy ciąg liczbowo-znakowy pozwalający zidentyfikować
zgłoszenie. ';

comment on column ZGLOSZENIE_PUNKTU_WYJSCIA.nr_pw is
'Jest to unikatowy ciąg liczbowo-znakowy punktu wyjścia, pozwalający go
scharakteryzować. ';

/*=====*/
/* Index: ZGLOSZENIE_PUNKTU_WYJSCIA_PK */
/*=====*/
create unique index ZGLOSZENIE_PUNKTU_WYJSCIA_PK on
ZGLOSZENIE_PUNKTU_WYJSCIA (
nr_zgloszenia ASC,
nr_pw ASC
);

/*=====*/
/* Index: ZGLOSZENIE_PUNKTU_WYJSCIA_FK */
/*=====*/
create index ZGLOSZENIE_PUNKTU_WYJSCIA_FK on ZGLOSZENIE_PUNKTU_WYJSCIA (
nr_zgloszenia ASC
);

/*=====*/
/* Index: ZGLOSZENIE_PUNKTU_WYJSCIA2_FK */
/*=====*/
create index ZGLOSZENIE_PUNKTU_WYJSCIA2_FK on ZGLOSZENIE_PUNKTU_WYJSCIA (
nr_pw ASC
);

alter table FAKTURA
add constraint FK_FAKTURA_FAKTURA_J_JST foreign key (nip_jst)
references JST (nip_jst)
on update restrict
on delete restrict;

comment on foreign key FAKTURA.FK_FAKTURA_FAKTURA_J_JST is
'Jest to związek łączący encję FAKTURA i JST. JST może posiadać wiele
faktur, a faktura może mieć tylko jedno JST. ';

alter table FAKTURA
add constraint FK_FAKTURA_OPIEKUN_F_OPIEKUN foreign key (id_opiekuna)
references OPIEKUN (id_opiekuna)
on update restrict
on delete restrict;

comment on foreign key FAKTURA.FK_FAKTURA_OPIEKUN_F_OPIEKUN is
'Jest to związek łączący encję FAKTURA i OPIEKUN. Opiekun może opiekować
się wieloma fakturami, a faktura może mieć tylko jednego opiekuna. ';

alter table GRUPA_ELEKTRYCZNA
add constraint FK_GRUPA_EL_OPIEKUN_G_OPIEKUN foreign key (id_opiekuna)

```

```

references OPIEKUN (id_opiekuna)
on update restrict
on delete restrict;

comment on foreign key GRUPA_ELEKTRYCZNA.FK_GRUPA_EL_OPIEKUN_G_OPIEKUN is
'Jest to związek łączący encję OPIEKUN i GRUPA_ELEKTRYCZNA. Opiekun może
opiekować się wieloma grupami elektrycznymi, a grupa elektryczna może mieć
tylko jednego opiekuna. ';

alter table GRUPA_GAZOWA
add constraint FK_GRUPA_GA_OPIEKUN_G_OPIEKUN foreign key (id_opiekuna)
references OPIEKUN (id_opiekuna)
on update restrict
on delete restrict;

comment on foreign key GRUPA_GAZOWA.FK_GRUPA_GA_OPIEKUN_G_OPIEKUN is
'Jest to związek łączący encję OPIEKUN i GRUPA_GAZOWA. Opiekun może
opiekować się wieloma grupami gazowymi, a grupa gazowa może mieć tylko
jednego opiekuna. ';

alter table OFERTA_ENERGIA
add constraint FK_OFERTA_E_OFERTA_EN_GRUPA_EL foreign key
(nr_grupy_elektrycznej)
references GRUPA_ELEKTRYCZNA (nr_grupy_elektrycznej)
on update restrict
on delete restrict;

comment on foreign key OFERTA_ENERGIA.FK_OFERTA_E_OFERTA_EN_GRUPA_EL is
'Jest to związek łączący encję OFERTA_ENERGIA i GRUPA_ELEKTRYCZNA. Grupa
elektryczna może otrzymać wiele ofert, a oferta jest skierowana do jednej,
konkretnej grupy elektrycznej.';

alter table OFERTA_ENERGIA
add constraint FK_OFERTA_E_OFERTA_EN_SPRZEDAW foreign key
(nip_sprzedawcy)
references SPRZEDAWCA (nip_sprzedawcy)
on update restrict
on delete restrict;

comment on foreign key OFERTA_ENERGIA.FK_OFERTA_E_OFERTA_EN_SPRZEDAW is
'Jest to związek łączący encję SPRZEDAWCA i OFERTA_ENERGIA. Sprzedawca
może złożyć wiele ofert, oferta może dotyczyć tylko jednego, konkretnego
sprzedawcy';

alter table OFERTA_ENERGIA
add constraint FK_OFERTA_E_PLATNOSC__PLATNOSC foreign key
(id_platnosci)
references PLATNOSC (id_platnosci)
on update restrict
on delete restrict;

comment on foreign key OFERTA_ENERGIA.FK_OFERTA_E_PLATNOSC__PLATNOSC is
'Jest to związek łączący encję OFERTA_ENERGIA i PLATNOSC. Płatność może
dotyczyć tylko jednej oferty, a oferta może mieć tylko jedną płatność. ';

```

```

alter table OFERTA_GAZ
    add constraint FK_OFERTA_G_OFERTA_GA_GRUPA_GA foreign key
(nr_grupy_gazowej)
    references GRUPA_GAZOWA (nr_grupy_gazowej)
    on update restrict
    on delete restrict;

comment on foreign key OFERTA_GAZ.FK_OFERTA_G_OFERTA_GA_GRUPA_GA is
'Jest to związek łączący encję OFERTA_GAZ i GRUPA_GAZOWA. Grupa gazowa
może otrzymać wiele ofert, a oferta jest skierowana do jednej, konkretnej
grupy gazowej.';

alter table OFERTA_GAZ
    add constraint FK_OFERTA_G_OFERTA_GA_SPRZEDAW foreign key
(nip_sprzedawcy)
    references SPRZEDAWCA (nip_sprzedawcy)
    on update restrict
    on delete restrict;

comment on foreign key OFERTA_GAZ.FK_OFERTA_G_OFERTA_GA_SPRZEDAW is
'Jest to związek łączący encję SPRZEDAWCA i OFERTA_GAZ. Sprzedawca może
złożyć wiele ofert, oferta może dotyczyć tylko jednego, konkretnego
sprzedawcy';

alter table OFERTA_GAZ
    add constraint FK_OFERTA_G_PLATNOSC__PLATNOSC foreign key
(id_platnosci_gaz)
    references PLATNOSC_GAZ (id_platnosci_gaz)
    on update restrict
    on delete restrict;

comment on foreign key OFERTA_GAZ.FK_OFERTA_G_PLATNOSC__PLATNOSC is
'Jest to związek łączący encję OFERTA_GAZ i PLATNOSC_GAZ. Płatność może
dotyczyć tylko jednej oferty, a oferta może mieć tylko jedną płatność.';

alter table PLATNOSC
    add constraint FK_PLATNOSC_PLATNOSC__OFERTA_E foreign key
(nr_oferty_energia)
    references OFERTA_ENERGIA (nr_oferty_energia)
    on update restrict
    on delete restrict;

comment on foreign key PLATNOSC.FK_PLATNOSC_PLATNOSC__OFERTA_E is
'Jest to związek łączący encję OFERTA_ENERGIA i PLATNOSC. Płatność może
dotyczyć tylko jednej oferty, a oferta może mieć tylko jedną płatność.';

alter table PLATNOSC_GAZ
    add constraint FK_PLATNOSC_PLATNOSC__OFERTA_G foreign key
(nr_oferty_gaz)
    references OFERTA_GAZ (nr_oferty_gaz)
    on update restrict
    on delete restrict;

```

```

comment on foreign key PLATNOSC_GAZ.FK_PLATNOSC_PLATNOSC__OFERTA_G is
'Jest to związek łączący encję OFERTA_GAZ i PLATNOSC_GAZ. Płatność może
dotyczyć tylko jednej oferty, a oferta może mieć tylko jedną płatność. ';

alter table POZYCJA
  add constraint FK_POZYCJA_POZYCJA_FAKTURA foreign key (id_faktury)
  references FAKTURA (id_faktury)
  on update restrict
  on delete restrict;

alter table POZYCJA
  add constraint FK_POZYCJA_POZYCJA2_WERSJA_P foreign key (id_wersji)
  references WERSJA_PRODUKTU (id_wersji)
  on update restrict
  on delete restrict;

alter table PPE
  add constraint FK_PPE_PPE_JST_JST foreign key (nip_jst)
  references JST (nip_jst)
  on update restrict
  on delete restrict;

comment on foreign key PPE.FK_PPE_PPE_JST_JST is
'Jest to związek łączący encję JST i PPE. JST może posiadać wiele PPE, a
PPE może należeć tylko do jednej JST.';

alter table PPE_GRUPY_ELEKTRYCZNEJ
  add constraint FK_PPE_GRP_PPE_GRUPY_PPE foreign key (nr_ppe)
  references PPE (nr_ppe)
  on update restrict
  on delete restrict;

comment on foreign key PPE_GRUPY_ELEKTRYCZNEJ.FK_PPE_GRP_PPE_GRUPY_PPE is
'Jest to związek łączący encję PPE i GRUPA_ELEKTRYCZNA. Grupa elektryczna
może posiadać wiele PPE, a PPE może znajdować się w wielu grupach
elektrycznych.';

alter table PPE_GRUPY_ELEKTRYCZNEJ
  add constraint FK_PPE_GRP_PPE_GRUPY_GRP_EL foreign key
(nr_grupy_elektrycznej)
  references GRUPA_ELEKTRYCZNA (nr_grupy_elektrycznej)
  on update restrict
  on delete restrict;

comment on foreign key
PPE_GRUPY_ELEKTRYCZNEJ.FK_PPE_GRP_PPE_GRUPY_GRP_EL is
'Jest to związek łączący encję PPE i GRUPA_ELEKTRYCZNA. Grupa elektryczna
może posiadać wiele PPE, a PPE może znajdować się w wielu grupach
elektrycznych.';

alter table PUNKT_WYJSCIA
  add constraint FK_PUNKT_WY_PUNKTY_WY_JST foreign key (nip_jst)
  references JST (nip_jst)
  on update restrict

```

```

on delete restrict;

comment on foreign key PUNKT_WYJSCIA.FK_PUNKT_WY_PUNKTY_WY_JST is
'Jest to związek łączący encję JST i PUNKT_WYJSCIA. JST może posiadać
wiele punktów wyjścia, a punkt wyjścia może należeć tylko do jednej JST.';

alter table PUNKT_WYJSCIA_GRUPY_GAZOWEJ
add constraint FK_PUNKT_WY_PUNKT_WYJ_PUNKT_WY foreign key (nr_pw)
references PUNKT_WYJSCIA (nr_pw)
on update restrict
on delete restrict;

comment on foreign key
PUNKT_WYJSCIA_GRUPY_GAZOWEJ.FK_PUNKT_WY_PUNKT_WYJ_PUNKT_WY is
'Jest to związek łączący encję PUNKT_WYJSCIA i GRUPA_GAZOWA. Grupa gazowa
może posiadać wiele punktów wyjścia, a punkt wyjścia może znajdować się w
wielu grupach gazowych.';

alter table PUNKT_WYJSCIA_GRUPY_GAZOWEJ
add constraint FK_PUNKT_WY_PUNKT_WYJ_GRUPA_GA foreign key
(nr_grupy_gazowej)
references GRUPA_GAZOWA (nr_grupy_gazowej)
on update restrict
on delete restrict;

comment on foreign key
PUNKT_WYJSCIA_GRUPY_GAZOWEJ.FK_PUNKT_WY_PUNKT_WYJ_GRUPA_GA is
'Jest to związek łączący encję PUNKT_WYJSCIA i GRUPA_GAZOWA. Grupa gazowa
może posiadać wiele punktów wyjścia, a punkt wyjścia może znajdować się w
wielu grupach gazowych.';

alter table UMOWA_WSP
add constraint FK_UMOWA_WS_OPIEKUN_U_OPIEKUN foreign key (id_opiekuna)
references OPIEKUN (id_opiekuna)
on update restrict
on delete restrict;

comment on foreign key UMOWA_WSP.FK_UMOWA_WS_OPIEKUN_U_OPIEKUN is
'Jest to związek łączący encję UMOWA_WSP i OPIEKUN. Opiekun może opiekować
się wieloma umowami, a umowa współpracy może mieć tylko jednego opiekuna.';

alter table UMOWA_WSP
add constraint FK_UMOWA_WS_UMOWA_WSP_JST foreign key (nip_jst)
references JST (nip_jst)
on update restrict
on delete restrict;

comment on foreign key UMOWA_WSP.FK_UMOWA_WS_UMOWA_WSP_JST is
'Jest to związek łączący encję JST i UMOWA_WSP. JST może posiadać wiele
umów współpracy, a umowa współpracy może dotyczyć tylko jednej JST.';

alter table WERSJA_PRODUKTU
add constraint FK_WERSJA_P_WERSJA_PR_PRODUKT foreign key (id_produktu)

```



```

references PRODUKT (id_produktu)
on update restrict
on delete restrict;

comment on foreign key WERSJA_PRODUKTU.FK_WERSJA_P_WERSJA_PR_PRODUKT is
'Jest to związek łączący encję PRODUKT i WERSJA_PRODUKTU. Produkt musi
posiadać jedną lub wiele wersji, a wersja może dotyczyć się tylko jednego
produktu. ';

alter table ZGLOSZENIE
add constraint FK_ZGLOSZEN_ZGLOSZENI_UMOWA_WS foreign key
(nr_umowy_wsp)
references UMOWA_WSP (nr_umowy_wsp)
on update restrict
on delete restrict;

comment on foreign key ZGLOSZENIE.FK_ZGLOSZEN_ZGLOSZENI_UMOWA_WS is
'Jest to związek łączący encję ZGLOSZENIE i UMOWA_WSP. UMOWA_WSP może
posiadać wiele wiele zgłoszeń, a zgłoszenie może dotyczyć tylko jednej
umowy współpracy.';

alter table ZGLOSZENIE_PPE
add constraint FK_ZGLOSZEN_ZGLOSZENI_PPE foreign key (nr_ppe)
references PPE (nr_ppe)
on update restrict
on delete restrict;

comment on foreign key ZGLOSZENIE_PPE.FK_ZGLOSZEN_ZGLOSZENI_PPE is
'Jest to związek łączący encję ZGLOSZENIE i PPE. Zgłoszenie może posiadać
wiele PPE, a PPE może znajdować się w wielu zgłoszeniach.';

alter table ZGLOSZENIE_PPE
add constraint FK_ZGLOSZEN_ZGLOSZENI_ZGLOSZEN foreign key
(nr_zgloszenia)
references ZGLOSZENIE (nr_zgloszenia)
on update restrict
on delete restrict;

comment on foreign key ZGLOSZENIE_PPE.FK_ZGLOSZEN_ZGLOSZENI_ZGLOSZEN is
'Jest to związek łączący encję ZGLOSZENIE i PPE. Zgłoszenie może posiadać
wiele PPE, a PPE może znajdować się w wielu zgłoszeniach.';

alter table ZGLOSZENIE_PUNKTU_WYJSCIA
add constraint FK_ZGLOSZEN_ZGLOSZENI_ZGLOSZEN foreign key
(nr_zgloszenia)
references ZGLOSZENIE (nr_zgloszenia)
on update restrict
on delete restrict;

comment on foreign key
ZGLOSZENIE_PUNKTU_WYJSCIA.FK_ZGLOSZEN_ZGLOSZENI_ZGLOSZEN is
'Jest to związek łączący encję ZGLOSZENIE i PUNKT_WYJSCIA. Zgłoszenie może
posiadać wiele punktów wyjścia, a punkt wyjścia może znajdować się w wielu
zgłoszeniach.';

```

```
alter table ZGLOSZENIE_PUNKTU_WYJSCIA
  add constraint FK_ZGLOSZEN_ZGLOSZENI_PUNKT_WY foreign key (nr_pw)
    references PUNKT_WYJSCIA (nr_pw)
    on update restrict
    on delete restrict;
```

comment on foreign key

ZGLOSZENIE_PUNKTU_WYJSCIA.FK_ZGLOSZEN_ZGLOSZENI_PUNKT_WY is

'Jest to związek łączący encję ZGLOSZENIE i PUNKT_WYJSCIA. Zgłoszenie może posiadać wiele punktów wyjścia, a punkt wyjścia może znajdować się w wielu zgłoszeniach.';

6. Widoki

a) Widok funkcyjny - v_punkty_do_zgloszenia

Widok przedstawia wykaz punktów zgłoszonych przez klientów, które jeszcze nie zostały zgłoszone do Sprzedawcy w celu za aneksowania umowy sprzedaży. Dodatkowo widok ukazuje numer NIP jednostki do którego należy punkt oraz nazwę Sprzedawcy i numer telefonu do którego należy dokonać zgłoszenia.

```
ALTER VIEW "DBA"."v_punkty_do_zgloszenia" AS
SELECT
    U.nip_jst,
    T1.nr_pw,
    Z.typ,
    T1.nr_zgloszenia,
    S.nazwa_sprzedawcy,
    S.nr_telefonu_sprzedawcy
FROM
(
    SELECT
        nr_pw,
        nr_zgloszenia
    FROM
        ZGLOSZENIE_PUNKTU_WYJSCIA
    UNION
    ALL
    SELECT
        nr_ppe,
        nr_zgloszenia
    FROM
        ZGLOSZENIE_PPE
) AS T1
JOIN ZGLOSZENIE AS Z ON Z.nr_zgloszenia = T1.nr_zgloszenia
JOIN UMOWA_WSP AS U ON U.nr_umowy_wsp = Z.nr_umowy_wsp
JOIN OPIEKUN AS O ON O.id_opiekuna = U.id_opiekuna
JOIN (
    SELECT
        nr_grupy_elektrycznej,
        id_opiekuna,
        najkorzystniejsza_stawka
    FROM
        GRUPA_ELEKTRYCZNA
    UNION
    ALL
    SELECT
        nr_grupy_gazowej,
        id_opiekuna,
        najkorzystniejsza_stawka
    FROM
```

```

        GRUPA_GAZOWA
    ) AS T2 ON T2.id_opiekuna = O.id_opiekuna
JOIN (
    SELECT
        nr_oferty_energia,
        nip_sprzedawcy,
        oferowana_stawka,
        nr_grupy_elektrycznej
    FROM
        OFERTA_ENERGIA
    UNION
    ALL
    SELECT
        nr_oferty_gaz,
        nip_sprzedawcy,
        oferowana_stawka,
        nr_grupy_gazowej
    FROM
        OFERTA_GAZ
) AS T3 ON T3.nr_grupy_elektrycznej = T2.nr_grupy_elektrycznej
AND T3.oferowana_stawka = T2.najkorzystniejsza_stawka
JOIN SPRZEDAWCA AS S ON S.nip_sprzedawcy = T3.nip_sprzedawcy
WHERE
    Z.status = 'Oczekujące'
ORDER BY
    S.nazwa_sprzedawcy DESC

```

Widok:

nip_jst	nr_pw	typ	nr_zgloszenia	nazwa_sprzedawcy	nr_telefonu_sprzedawcy
11250037105	277932495688	USUNIECIE	15	Fortum Power and Heat Polska Sp. Z o.o.	609457321
21250037105	307300225725	USUNIECIE	15	Fortum Power and Heat Polska Sp. Z o.o.	609457321
31250037105	364746959200	USUNIECIE	15	Fortum Power and Heat Polska Sp. Z o.o.	609457321
45521567404	224329093618	DODANIE	1	Enea S.A.	517933477

b) Widok funkcyjny - v_najlepszy_zespol

Widok wyłania zespół, który dokonał największego obrotu z umów współpracy w roku poprzednim. Widok zwróci dwa zespoły w przypadku uzyskania przez nie wyniku ex aequo. Poza zespołem i wykonanym obrotem, zostanie wyświetlona również liczność zwycięskiego zespołu.

```
ALTER VIEW "DBA"."v_najlepszy_zespol" AS

SELECT
    SUBSTRING(O.id_opiekuna, 1, 1) AS ZESPOL,
    COUNT(DISTINCT O.id_opiekuna) AS LICZNOSC_ZESPOLU,
    SUM(U.kwota) AS OBROT
FROM
    OPIEKUN AS O
    LEFT JOIN UMOWA_WSP AS U ON U.id_opiekuna = O.id_opiekuna
WHERE
    ((date(NOW(*))) - (U.data_zawarcia)) < 365
GROUP BY
    SUBSTRING(O.id_opiekuna, 1, 1)
HAVING
    SUM(U.kwota) =(
        SELECT
            TOP 1 SUM(U.kwota) AS OBROT
        FROM
            OPIEKUN AS O
            LEFT JOIN UMOWA_WSP AS U ON U.id_opiekuna = O.id_opiekuna
        WHERE
            ((date(NOW(*))) - (U.data_zawarcia)) < 365
        GROUP BY
            SUBSTRING(O.id_opiekuna, 1, 1)
        ORDER BY
            OBROT DESC
    )
)
```

Widok:

	ZESPOL	LICZNOSC_ZESPOLU	OBROT
1	A	1	79 000
2	D	3	79 000

c) Widok zmaterializowany - vm_wynik_roczny

Widok przedstawia numer identyfikacyjny opiekuna, imię oraz nazwisko opiekuna wraz ilości umów podpisanych w poprzednim roku oraz ilością grup zakupowych, których był opiekunem.

```
ALTER VIEW "DBA"."vm_wynik_roczny" AS
SELECT
    T1.id_opiekuna,
    T1.Imie,
    T1.Nazwisko,
    T1.l_umow,
    COUNT(T2.id_opiekuna) AS l_grup
FROM
    (
        SELECT
            O.id_opiekuna,
            O.Imie,
            O.Nazwisko,
            COUNT(U.nr_umowy_wsp) AS l_umow
        FROM
            OPIEKUN AS O
        LEFT JOIN UMOWA_WSP AS U ON U.id_opiekuna = O.id_opiekuna
        WHERE
            YEAR(date(NOW(*))) - YEAR(U.data_zawarcia) = 1
            AND U.kwota > 2000
        GROUP BY
            O.id_opiekuna,
            O.Imie,
            O.Nazwisko
        ORDER BY
            l_umow DESC
    ) AS T1
LEFT JOIN(
    SELECT
        nr_grupy_elektrycznej,
        id_opiekuna
    FROM
        GRUPA_ELEKTRYCZNA
    WHERE
        YEAR(date(NOW(*))) - YEAR(data_rozstrzygniecie) = 1
    UNION
    ALL
    SELECT
        nr_grupy_gazowej,
        id_opiekuna
    FROM
        GRUPA_GAZOWA
    WHERE
        YEAR(date(NOW(*))) - YEAR(data_rozstrzygniecie) = 1
```

```

    ) AS T2 ON T2.id_opiekuna = T1.id_opiekuna
GROUP BY
    T1.id_opiekuna,
    T1.Imie,
    T1.Nazwisko,
    T1.l_umow
ORDER BY
    T1.l_umow DESC,
    l_grup DESC

```

Widok:

	id_opiekuna	Imie	Nazwisko	l_umow	l_grup
1	D08	Patryk	Płachciński	8	1
2	A01	Emilia	Celińska	7	2
3	C10	Sebastian	Morawietz	7	0
4	D12	Urszula	Niewęglowska	5	1
5	E15	Tomasz	Sadowski	5	0
6	E17	Łukasz	Stolarczyk	5	0
7	C07	Aleksandra	Grzegorzółka	4	0
8	B05	Sebastian	Madej	3	0
9	C06	Krzysztof	Maciuszko	3	0
10	B03	Estera	Zajęc	2	1
11	B13	Przemysław	Jaworski	2	0
12	D09	Wojciech	Kaczmarek	1	0
13	E16	Roman	Bartyzel	1	0

7. Procedury

a) Procedura z użyciem kursora - obnizka_ceny

Procedura sprawdza dochody z ostatniego roku każdego produktu. W przypadku dokonania obrotu przez produkt mniejszego niż 10000 złotych, dokonuje zmiany ceny produktu – obniża ją o 10 procent przy pomocy dodania rekordu do tabeli WERSJA_PRODUKTU.

```

ALTER PROCEDURE "DBA"."obnizka_ceny"(IN _procent float(2)) BEGIN
DECLARE _id_produktu char(10);

DECLARE _cena int;

DECLARE _ilosc int;

DECLARE _pozycja int;

DECLARE _dochod float(2);

DECLARE _kursor CURSOR FOR (
    SELECT
        P.id_produktu,
        P.cena,
        SUM(P0.ilosc)

```

```

FROM
    Produkt AS P
    JOIN WERSJA_PRODUKTU AS W ON W.id_produktu = P.id_produktu
    AND W.nowa_cena = P.cena
    LEFT JOIN POZYCJA AS PO ON PO.id_wersji = W.id_wersji
GROUP BY
    P.id_produktu,
    P.cena,
    W.data_zmiany
);

SET
    _procent = _procent / 100;

OPEN _kursor;

_kolko: LOOP FETCH NEXT _kursor INTO _id_produktu,
    _cena,
    _ilosc;

IF @@sqlstatus != 0 THEN
    LEAVE _kolko;
END IF;

SET
    _dochod = (_ilosc * _cena);

IF _dochod < 10000
OR _dochod IS NULL THEN
    INSERT INTO
        WERSJA_PRODUKTU (id_wersji, id_produktu, data_zmiany, nowa_cena)
    VALUES
        (
            WERSJA_PRODUKT_seq.NEXTVAL,
            _id_produktu,
            date(NOW(*)),
            _procent * _cena
        );

    MESSAGE 'Zmieniam cene produktu o id:' || _id_produktu TO CLIENT;

END IF;

END LOOP;
CLOSE _kursor;
END

```


Działanie:

Przed wywołaniem procedury:

	id_produktu	nazwa_produktu	cena
1	1	Philips Solar	360
2	10	Philips LED	280
3	2	Optica	780
4	3	Harlem LUM	560
5	4	Area 4500	280
6	8	Pike JDOB	560

Wywołanie procedury:

```
1 call obnizka_ceny()
```

<

Results

Zmieniam cene produktu o id:10
Zmieniam cene produktu o id:3
Zmieniam cene produktu o id:4
Zmieniam cene produktu o id:8
1 row(s) affected
Execution time: 0.022 seconds
Procedure completed

Rezultat po wykonaniu procedury:

	id_wersji	id_produktu	data_zmiany	nowa_cena
1	1	1	2022-01-03	560
2	2	2	2022-01-10	780
3	3	3	2022-01-10	590
4	4	4	2021-11-15	280
5	5	8	2022-01-02	550
6	6	10	2021-08-15	750
7	580269	10	2022-01-26	675
8	580270	3	2022-01-26	531
9	580271	4	2022-01-26	252
10	580272	8	2022-01-26	495

b) Procedura z użyciem transakcji - aktualizacja_ppe

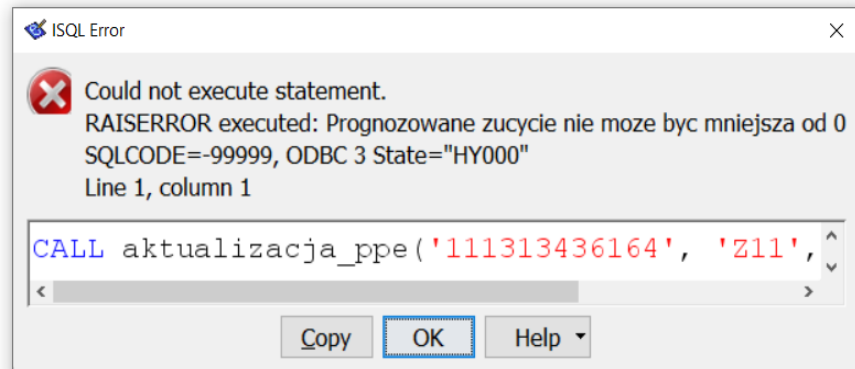
Procedura aktualizuje tabelę PPE jednocześnie pilnując aby wprowadzona taryfa była prawidłowa, prognozowane zużycie było większe od zera, a rodzaj umowy kompleksowy lub rozdzielny.

```
ALTER PROCEDURE "DBA"."aktualizacja_ppe"(  
    IN _nr_ppe char(40),  
    IN _taryfa char(10),  
    IN _rodzaj_umowy char(20),  
    IN _prognozowane_zuzycie int  
) AS BEGIN DECLARE @p_nr_ppe char(40) DECLARE kursor CURSOR FOR (  
    SELECT  
        nr_ppe  
    FROM  
        PPE  
) OPEN kursor WHILE(1 = 1) BEGIN FETCH NEXT kursor INTO @p_nr_ppe IF  
@p_nr_ppe = _nr_ppe  
OR @ @sqlstatus != 0 BEGIN BREAK  
END  
END CLOSE kursor IF @ @sqlstatus != 0 BEGIN RAISERROR 99999 'PPE o  
podanym numerze nie istnieje'  
END BEGIN TRANSACTION  
UPDATE  
    PPE  
SET  
    taryfa = _taryfa,  
    rodzaj_umowy = _rodzaj_umowy,  
    prognozowane_zuzycie = _prognozowane_zuzycie  
WHERE  
    nr_ppe = _nr_ppe IF _taryfa NOT LIKE '[B-C]%'  
    AND _taryfa NOT LIKE 'G%' BEGIN RAISERROR 99999 'Nieprawidlowa  
rodzaj taryfy' ROLLBACK TRANSACTION  
END IF _rodzaj_umowy NOT LIKE 'Rozdzielna'  
AND _rodzaj_umowy NOT LIKE 'Kompleksowa' BEGIN RAISERROR 99999  
'Nieprawidlowa rodzaj umowy' ROLLBACK TRANSACTION  
END IF _prognozowane_zuzycie <= 0 BEGIN RAISERROR 99999 'Prognozowane  
zucycie nie moze byc mniejsza od 0' ROLLBACK TRANSACTION  
END  
ELSE BEGIN COMMIT TRANSACTION  
END  
END
```

Działanie:

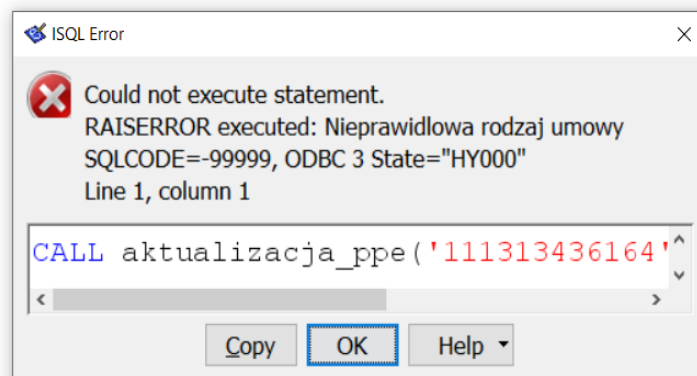
Błędna taryfa, rodzaj umowy oraz prognozowane zużycie:

```
CALL aktualizacja_ppe('111313436164', 'Z11', 'Okresowa', '-2')
```



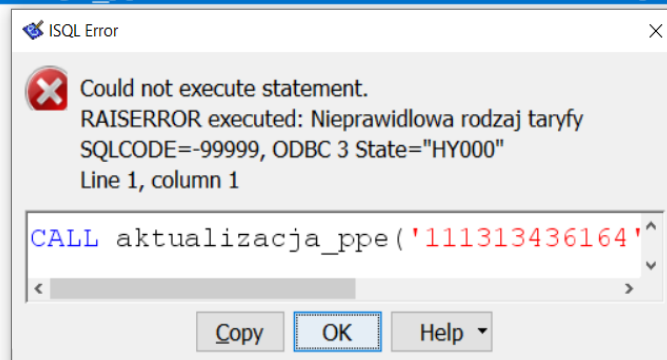
Błędna taryfa oraz rodzaj umowy:

```
CALL aktualizacja_ppe('111313436164', 'Z11', 'Okresowa', '450')
```



Błędna taryfa:

```
CALL aktualizacja_ppe('111313436164', 'Z11', 'Kompleksowa', '450')
```



c) **Procedura z użyciem transakcji - dodaj_pracownika**

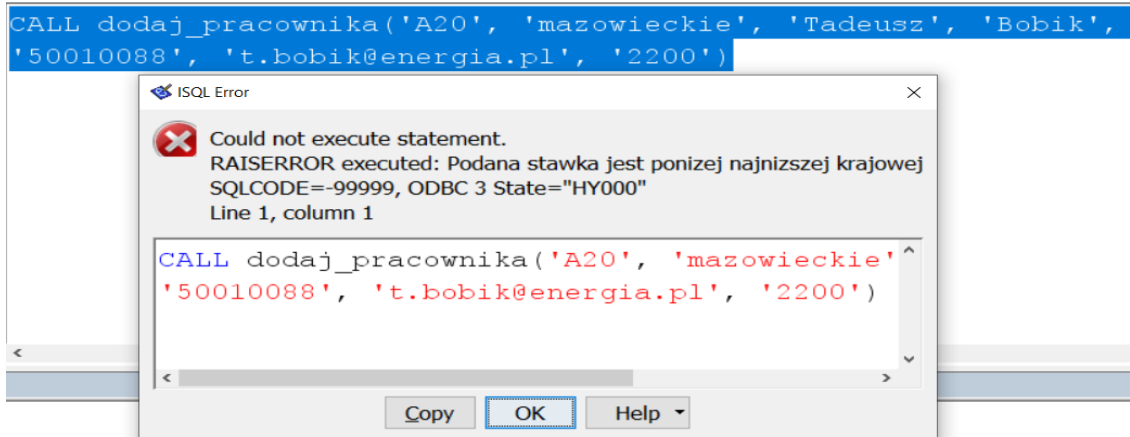
Procedura dodaje nowy rekord do tabeli OPIEKUN jednocześnie sprawdzając czy wprowadzone dane są prawidłowe. Numer telefonu musi posiadać dziewięć cyfr, mail służbowy musi kończyć się na @voltra.pl oraz podstawa wynagrodzenia nie może być poniżej płacy minimalnej.

```
ALTER PROCEDURE "DBA"."dodaj_pracownika"(  
    IN _id_opiekuna char(3),  
    IN _wojewodztwo char(20),  
    IN _imie char(50),  
    IN _nazwisko char(50),  
    IN _numer_telefonu char(9),  
    IN _email char(50),  
    IN _podstawa int  
) AS BEGIN DECLARE @p_id_opiekuna char(3) DECLARE kursor CURSOR FOR (  
    SELECT  
        id_opiekuna  
    FROM  
        OPIEKUN  
) OPEN kursor WHILE(1 = 1) BEGIN FETCH NEXT kursor INTO  
@p_id_opiekuna IF @p_id_opiekuna = _id_opiekuna  
OR @ @sqlstatus != 0 BEGIN BREAK  
END  
END CLOSE kursor IF @ @sqlstatus = 0 BEGIN RAISERROR 99999 'Podane ID  
jest juz zajete'  
END BEGIN TRANSACTION  
INSERT INTO  
    OPIEKUN (  
        id_opiekuna,  
        wojewodztwo,  
        imie,  
        nazwisko,  
        numer_telefonu,  
        data_zatrudnienia,  
        email_opiekun,  
        podstawa  
    )  
VALUES  
(  
    _id_opiekuna,  
    _wojewodztwo,  
    _imie,  
    _nazwisko,  
    _numer_telefonu,  
    NOW(*),  
    _email,  
    _podstawa  
) IF _numer_telefonu NOT LIKE '_____' BEGIN ROLLBACK  
TRANSACTION RAISERROR 99999 'Bledny numer telefonu'  
END IF _email NOT LIKE '%@voltra.pl' BEGIN ROLLBACK TRANSACTION  
RAISERROR 99999 'Bledny adres email'  
END IF _id_opiekuna NOT LIKE '[A-E]%' BEGIN ROLLBACK TRANSACTION  
RAISERROR 99999 'Podany zepol nie istnieje'
```

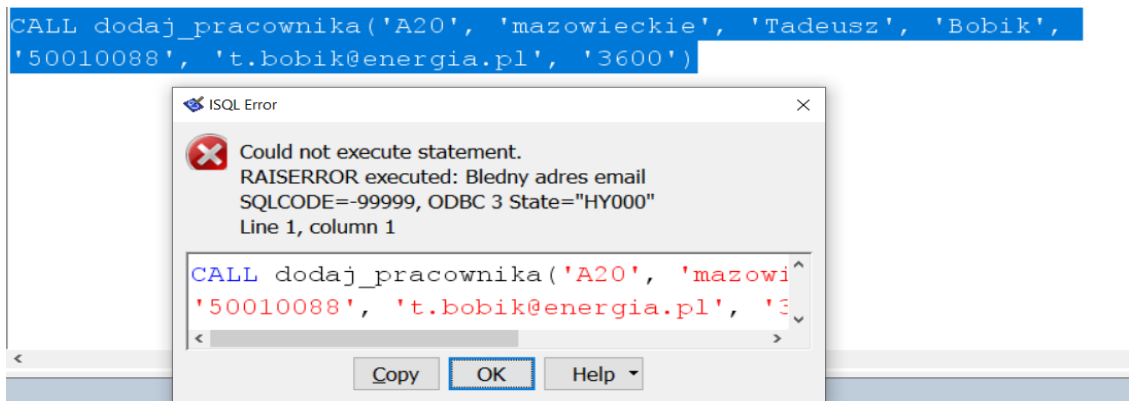
```
END IF _podstawa < 2800 BEGIN ROLLBACK TRANSACTION RAISERROR 99999  
'Podana stawka jest ponizej najnizszej krajowej'  
END  
ELSE BEGIN COMMIT TRANSACTION  
END  
END
```

Działanie:

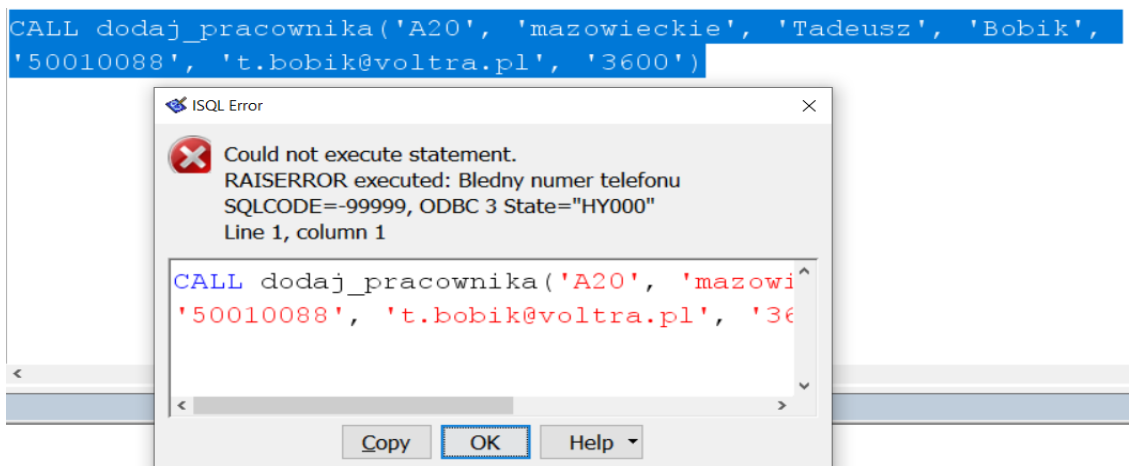
Błędny numer (zbyt krótki), błędny email (nie zawiera końcówki @voltra.pl), błędna wartość podstawy (poniżej najniższej krajowej):



Błędny numer (zbyt krótki), błędny email (nie zawiera końcówki @voltra.pl):



Błędny numer (zbyt krótki):



8. Funkcje

a) Funkcja z użyciem kursora - podwyżka

Funkcja podwyżka oblicza sugerowaną podwyżkę dla pracownika, uzależnioną od wysokości wykonanego obrotu. Jeżeli procentowy udział w obrocie całej firmy jest większy od jedenastu procent wówczas funkcja zwróci wartość podwyżki, w przeciwnym wypadku zwróci zero. Jako atrybut funkcja przyjmuje identyfikator pracownika.

```
ALTER FUNCTION "DBA"."podwyżka"(IN p_id_opiekuna char(3)) RETURNS
INTEGER DETERMINISTIC BEGIN DECLARE _podwyżka int;
DECLARE _id_opiekuna char(3);
DECLARE _suma int;
DECLARE _obrot int;
DECLARE _procent int;
DECLARE _kursor CURSOR FOR (
    SELECT
        O.id_opiekuna,
        SUM(U.kwota) AS suma
    FROM
        OPIEKUN AS O
        LEFT JOIN UMOWA_WSP AS U ON U.id_opiekuna = O.id_opiekuna
    WHERE
        YEAR(date(NOW(*))) - YEAR(U.data_zawarcia) = 1
    GROUP BY
        O.id_opiekuna);
SET
    _obrot = obrot();
SET
    _podwyżka = 0;
OPEN _kursor;
_petla: LOOP FETCH _kursor INTO _id_opiekuna,
    _suma;
IF _id_opiekuna = p_id_opiekuna
OR @@sqlstatus != 0 THEN
SET
    _suma = _suma * 100;
SET
    _procent = _suma / _obrot;
IF _procent >= 10 THEN
SET
    _procent = _procent / 2;
SET
    _podwyżka = _procent * 4000 / 100;
END IF;
LEAVE _petla;
END IF;
END LOOP;
IF @@sqlstatus != 0 THEN RAISERROR 99999 'Nie istnieje pracownik
o podanym ID'
END IF;
CLOSE _kursor;
RETURN _podwyżka;
END
```

Działanie:

1	SELECT podwyzka('A01')
2	
<	
Results	
podwyzka('A01')	
1	245

1	SELECT podwyzka('D12')
2	
<	
Results	
podwyzka('D12')	
1	0

b) Funkcja z użyciem kursora - obrot

Funkcja oblicza obrót firmy z umów współpracy za poprzedni rok.

```
ALTER FUNCTION "DBA"."obrot"() RETURNS INTEGER DETERMINISTIC BEGIN
DECLARE _obrot int;

SET
    _obrot =(
        SELECT
            SUM(U.kwota) AS OBROT
        FROM
            OPIEKUN AS O
            LEFT JOIN UMOWA_WSP AS U ON U.id_opiekuna = O.id_opiekuna
        WHERE
            (YEAR(date(NOW(*))) - YEAR(U.data_zawarcia)) = 1
    );

RETURN _obrot;

END
```

Działanie:

obrot()	
1	297 000

c) Funkcja z użyciem kursora - wartosc_faktury

Funkcja wartosc_faktury oblicza sumę wszystkich pozycji faktury, a następnie zwraca ją. Jako atrybut przyjmuje identyfikator faktury.

```
ALTER FUNCTION "DBA"."wartosc_faktury"(IN p_id_faktury int) RETURNS
INT DETERMINISTIC BEGIN DECLARE _wartosc_faktury int;

DECLARE _wartosc_faktury int;
DECLARE _id_faktury char(10);
DECLARE _koszt_zamowienia FLOAT(2);
DECLARE _kursor CURSOR FOR (
    SELECT
        T1.id_faktury,
        SUM(wartosc_pozycji) AS wartosc
    FROM
        (
            SELECT
                F.id_faktury,
                (P.ilosc * W.nowa_cena) AS wartosc_pozycji
            FROM
                FAKTURA AS F
                JOIN POZYCJA AS P ON P.id_faktury = F.id_faktury
                JOIN WERSJA_PRODUKTU AS W ON W.id_wersji =
P.id_wersji
        ) AS T1
    GROUP BY
        T1.id_faktury
);
OPEN _kursor;
_petla: LOOP FETCH _kursor INTO _id_faktury,
_wartosc_faktury;

SET
    _wartosc_faktury = _wartosc_faktury;

IF _id_faktury = p_id_faktury
OR @ @sqlstatus != 0 THEN LEAVE _petla;
END IF;
END LOOP;
IF @ @sqlstatus != 0 THEN RAISERROR 99999 'Faktura o podanym numerze
nie istnieje'
END IF;
CLOSE _kursor;
RETURN _wartosc_faktury;
END
```

Działanie:

1	SELECT wartosc_faktury('2')
Results	
wartosc_faktury('2')	
1	4 480

d) Funkcja z użyciem transakcji i kursora - odprawa_pracownika

Funkcja `odprawa_pracownika` sprawdza efektywność pracownika. W przypadku stwierdzenia zbyt małej ilości nowych umów zawartych w 365 ostatnich dni, ustalana jest data zwolnienia pracownika na 30 dni od dnia dzisiejszego. Dodatkowo funkcja zwraca wartość odprawy obliczanej na podstawie podpisanych umów w ciągu ostatnich 365 dni. Osoba mające cztery lub mniej umów nie otrzymują odprawy i funkcja zwraca wartość zero.

```
ALTER FUNCTION "DBA"."odprawa_pracownika"(@p_id_opiekuna char(3))
RETURNS int AS BEGIN DECLARE @odprawa int DECLARE @id_opiekuna
char(3) DECLARE @suma int DECLARE @kursor CURSOR FOR(
    SELECT
        O.id_opiekuna,
        COUNT(U.nr_umowy_wsp) AS l_umow
    FROM
        OPIEKUN AS O
        LEFT JOIN UMOWA_WSP AS U ON U.id_opiekuna = O.id_opiekuna
    WHERE
        (date(NOW(*))) -(U.data_zawarcia) < 365
        OR U.data_zawarcia IS NULL
    GROUP BY
        O.id_opiekuna
)
SET
    @odprawa = 0 OPEN @kursor WHILE(1 = 1) BEGIN FETCH @kursor INTO
    @id_opiekuna,
    @suma IF @id_opiekuna = @p_id_opiekuna
    OR @ @sqlstatus != 0 BEGIN BREAK
END
END CLOSE @kursor IF @ @sqlstatus != 0 BEGIN RAISERROR 99999 'Nie
istnieje pracownik o podanym ID'
END BEGIN TRANSACTION tr
UPDATE
    OPIEKUN
SET
    data_zwolnienia = NOW(*) + 90
WHERE
    id_opiekuna = @p_id_opiekuna
SET
    @odprawa = (@suma * 130) + 50 IF @suma > 4 ROLLBACK TRANSACTION
tr
ELSE
SET
    @odprawa = 0 COMMIT TRANSACTION tr RETURN @odprawa
END
```

Działanie:

1	SELECT	odprawa_pracownika('A01')
<		
Results		
odprawa_pracownika('A01')		
1		960

e) Funkcja z użyciem kursora - wielkość_wadium

Funkcja wielkość_wadium zwraca wielkość wadium jaką powinien wpłacić oferent aby jego oferta była brana pod uwagę podczas wyłaniania najkorzystniejszej oferty. Jako atrybut funkcja przyjmuje numer grupy elektrycznej, której dotyczy oferta.

```
ALTER FUNCTION "DBA"."wielkosc_wadium"(IN p_nr_oferty_energia
char(10)) RETURNS FLOAT DETERMINISTIC BEGIN DECLARE _wadium float;
DECLARE _nr_oferty_energia char(10);
DECLARE _kursor CURSOR FOR (
    SELECT
        O.nr_oferty_energia,
        G.wadium
    FROM
        GRUPA_ELEKTRYCZNA AS G
        JOIN OFERTA_ENERGIA AS O ON O.nr_grupy_elektrycznej =
G.nr_grupy_elektrycznej
);
OPEN _kursor;
_petla: LOOP FETCH _kursor INTO _nr_oferty_energia,
_wadium;
IF _nr_oferty_energia = p_nr_oferty_energia
OR @ @sqlstatus != 0 THEN LEAVE _petla;
END IF;
END LOOP;
CLOSE _kursor;
IF @ @sqlstatus != 0 THEN RAISERROR 99999 'Oferta o podanym numerze
nie istnieje'
END IF;
RETURN _wadium;
END
```

Działanie:

1	SELECT	wielkosc_wadium('1')
<		
Results		
wielkosc_wadium('1')		
1		6 506 947

f) Funkcja z użyciem kursora - obecna_stawka

Funkcja `obecna_stawka` zwraca najkorzystniejszą zaoferowaną stawkę grupy elektrycznej. Jako atrybut przyjmuje numer grupy elektrycznej.

```
ALTER FUNCTION "DBA"."obecna_stawka"(IN p_nr_grupy_elektrycznej
char(10)) RETURNS INTEGER DETERMINISTIC BEGIN DECLARE _stawka int;
DECLARE _nr_grupy_elektrycznej char(10);
DECLARE _kursor CURSOR FOR (
    SELECT
        nr_grupy_elektrycznej,
        najkorzystniejsza_stawka
    FROM
        GRUPA_ELEKTRYCZNA
);
OPEN _kursor;
_petla: LOOP FETCH _kursor INTO _nr_grupy_elektrycznej,
_stawka;

IF _nr_grupy_elektrycznej = p_nr_grupy_elektrycznej
OR @@sqlstatus != 0 THEN LEAVE _petla;

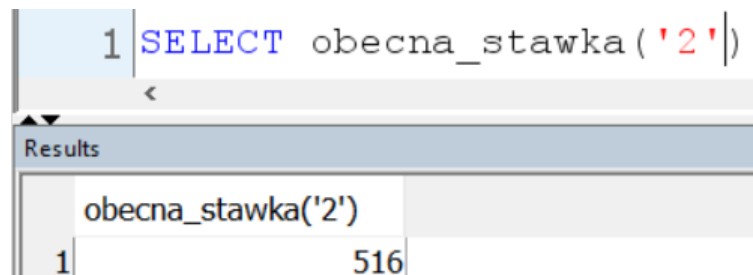
END IF;

END LOOP;

IF @@sqlstatus != 0 THEN RAISERROR 99999 'Podany numer grupy nie
istnieje'
END IF;

CLOSE _kursor;
RETURN _stawka;
END
```

Działanie:



The screenshot shows a SQL query editor with the command `SELECT obecna_stawka('2')` entered. Below the editor, a 'Results' window displays the output of the function call. The result is shown in a table with one row and one column, where the column header is `obecna_stawka('2')` and the value is `516`.

obecna_stawka('2')
516

9. Wyzwalacze (triggery)

a) Wyzwalacz – aktualizacja_wolumenu

Wyzwalacz aktualizacja_wolumenu po przypisaniu/usunięciu/aktualizacji punktu poboru energii elektrycznej wobec grupy elektrycznej aktualizuje wolumen grupy oraz wysokość wadium.

```
ALTER TRIGGER "aktualizacja_wolumenu"
AFTER
INSERT,
DELETE,
UPDATE
ORDER 1 ON "DBA"."PPE_GRUPY_ELEKTRYCZNEJ" REFERENCING NEW AS
new_name FOR EACH ROW BEGIN
UPDATE
GRUPA_ELEKTRYCZNA
SET
wolumen_energii = (
SELECT
SUM(PP.prognozowane_zuzycie)
FROM
PPE_GRUPY_ELEKTRYCZNEJ AS P
JOIN PPE AS PP ON PP.nr_ppe = P.nr_ppe
GROUP BY
P.nr_grupy_elektrycznej
HAVING
P.nr_grupy_elektrycznej = new_name.nr_grupy_elektrycznej
)
WHERE
nr_grupy_elektrycznej = new_name.nr_grupy_elektrycznej;

UPDATE
GRUPA_ELEKTRYCZNA
SET
wadium = 884 * 0.024 *(
SELECT
SUM(PP.prognozowane_zuzycie)
FROM
PPE_GRUPY_ELEKTRYCZNEJ AS P
JOIN PPE AS PP ON PP.nr_ppe = P.nr_ppe
GROUP BY
P.nr_grupy_elektrycznej
HAVING
P.nr_grupy_elektrycznej = new_name.nr_grupy_elektrycznej
)
WHERE
nr_grupy_elektrycznej = new_name.nr_grupy_elektrycznej;

END
```

Działanie:

Przed powiązaniem punktu z grupą:

	nr_grupy_elektrycznej	wolumen_energii	wadium
1	1	117 680	2 496 698
2	2	142 100	3 014 793
3	3	306 700	6 506 947
4	4	370 140	7 852 890
5	5	2	2

Po powiązaniu punktu z grupą numer jeden:

	nr_grupy_elektrycznej	wolumen_energii	wadium
1	1	117 700	2 497 123
2	2	142 100	3 014 793
3	3	306 700	6 506 947
4	4	370 140	7 852 890
5	5	2	2

b) Wyzwalacz - status_wadium

Wyzwalacz najkorzystniejsza_oferta po wpłynięciu wadium sprawdza czy wysokość wadium jest odpowiednia i w przypadku zbyt małej wartości ustawia w tabeli OFERTA_ENERGIA dla oferty, której dotyczyła płatność status_wadium jako NIEOPLACONE, a w przypadku odpowiedniej wysokości na OPLACONE.

```
ALTER TRIGGER "status_wadium" BEFORE
INSERT,
DELETE,
UPDATE
    ORDER 1 ON "DBA"."PLATNOSC" REFERENCING NEW AS new_name FOR EACH
ROW BEGIN DECLARE wpplacone_wadium int;
SET
    wpplacone_wadium = wielkosc_wadium(new_name.nr_oferty_energia);
UPDATE
    OFERTA_ENERGIA
SET
    status_wadium = 'OPLACONE'
WHERE
    nr_oferty_energia = new_name.nr_oferty_energia;
IF new_name.kwota < wpplacone_wadium THEN
UPDATE
    OFERTA_ENERGIA
SET
    status_wadium = 'NIEOPLACONE'
WHERE
    nr_oferty_energia = new_name.nr_oferty_energia;
MESSAGE 'The current date and time: ',
Now();
MESSAGE 'Opłata jest mniejsza niz wartosc zamowienia';
MESSAGE 'Opłata wynosi :',
new_name.kwota;
ENDIF;
END
```

Działanie:

Przed wpłynięciem wadium:

nr_oferty_energia	status_wadium
11	NIEOPLACONE
22	OPLACONE
33	OPLACONE
44	(NULL)
55	(NULL)
66	OPLACONE
77	OPLACONE
88	OPLACONE
99	OPLACONE
1011	(NULL)

Po wpłynięciu zbyt małego wadium dotyczącego czwartej oferty:

nr_oferty_energia	status_wadium
11	NIEOPLACONE
22	OPLACONE
33	OPLACONE
44	NIEOPLACONE
55	(NULL)
66	OPLACONE
77	OPLACONE
88	OPLACONE
99	OPLACONE
1011	(NULL)

c) **Wyzwalacz – najkorzystniejsza_oferta**

Wyzwalacz najkorzystniejsza_oferta po wpłynięciu płatności sprawdza czy oferowana stawka jest niższa od najkorzystniejszej stawki grupy elektrycznej. Jeżeli oferowana stawka jest niższa oraz status_wadium jest OPLACONE wówczas zmieniana jest najkorzystniejsza_stawka w tabeli GRUPA_ELEKTRYCZNA.

```
ALTER TRIGGER "najkorzystniejsza_oferta"
AFTER
INSERT,DELETE, UPDATE
ORDER 1 ON "DBA"."OFERTA_ENERGIA" REFERENCING OLD AS old_name NEW
AS new_name FOR EACH ROW BEGIN DECLARE _obecna_stawka int;
SET
    _obecna_stawka = obecna_stawka(new_name.nr_grupy_elektrycznej);
IF _obecna_stawka > new_name.oferowana_stawka
AND new_name.status_wadium = 'OPLACONE' THEN
UPDATE
    GRUPA_ELEKTRYCZNA
SET
    najkorzystniejsza_stawka = new_name.oferowana_stawka
WHERE
    nr_grupy_elektrycznej = new_name.nr_grupy_elektrycznej;
END IF;
```

d) Wyzwalacz – zmiana_ceny

Wyzwalacz zmiana_ceny po wprowadzeniu nowego rekordu do tabeli WERSJA_PRODUKTU zmienia wartość atrybutu cena w tabeli PRODUKT.

```
ALTER TRIGGER "zmiana_ceny"
AFTER
INSERT
ORDER 1 ON "DBA"."WERSJA_PRODUKTU" REFERENCING NEW AS new_name
FOR EACH ROW BEGIN
UPDATE
PRODUKT
SET
cena = (
SELECT
nowa_cena
FROM
WERSJA_PRODUKTU AS W
WHERE
W.id_wersji = new_name.id_wersji
)
WHERE
id_produktu = new_name.id_produktu;
END
```

Działanie:

Przed dodaniem nowej wersji:

1	SELECT *	1	SELECT *
2	FROM PRODUKT	2	FROM WERSJA_PRODUKTU
Results		Results	
id_produktu	nazwa_produktu	cena	
11	Philips Solar	560	
22	Optica	780	
33	Harlem LUM	590	
44	Area 4500	280	
58	Pike JDOB	550	
610	Philips LED	750	

id_wersji	id_produktu	data_zmiany	nowa_cena
11	1	2022-01-03	560
22	2	2022-01-10	780
33	3	2022-01-10	590
44	4	2021-11-15	280
55	8	2022-01-02	550
66	10	2021-08-15	750

Po dodaniu rekordu tabeli WERSJA_PRODUKTU:

1	SELECT *	1	SELECT *
2	FROM WERSJA_PRODUKTU	2	FROM PRODUKT
Results		Results	
id_wersji	id_produktu	data_zmiany	nowa_cena
11	1	2022-01-03	560
22	2	2022-01-10	780
33	3	2022-01-10	590
44	4	2021-11-15	280
55	8	2022-01-02	550
66	10	2021-08-15	750
77	10	2022-01-25	950

id_produktu	nazwa_produktu	cena
11	Philips Solar	560
22	Optica	780
33	Harlem LUM	590
44	Area 4500	280
58	Pike JDOB	550
610	Philips LED	950