

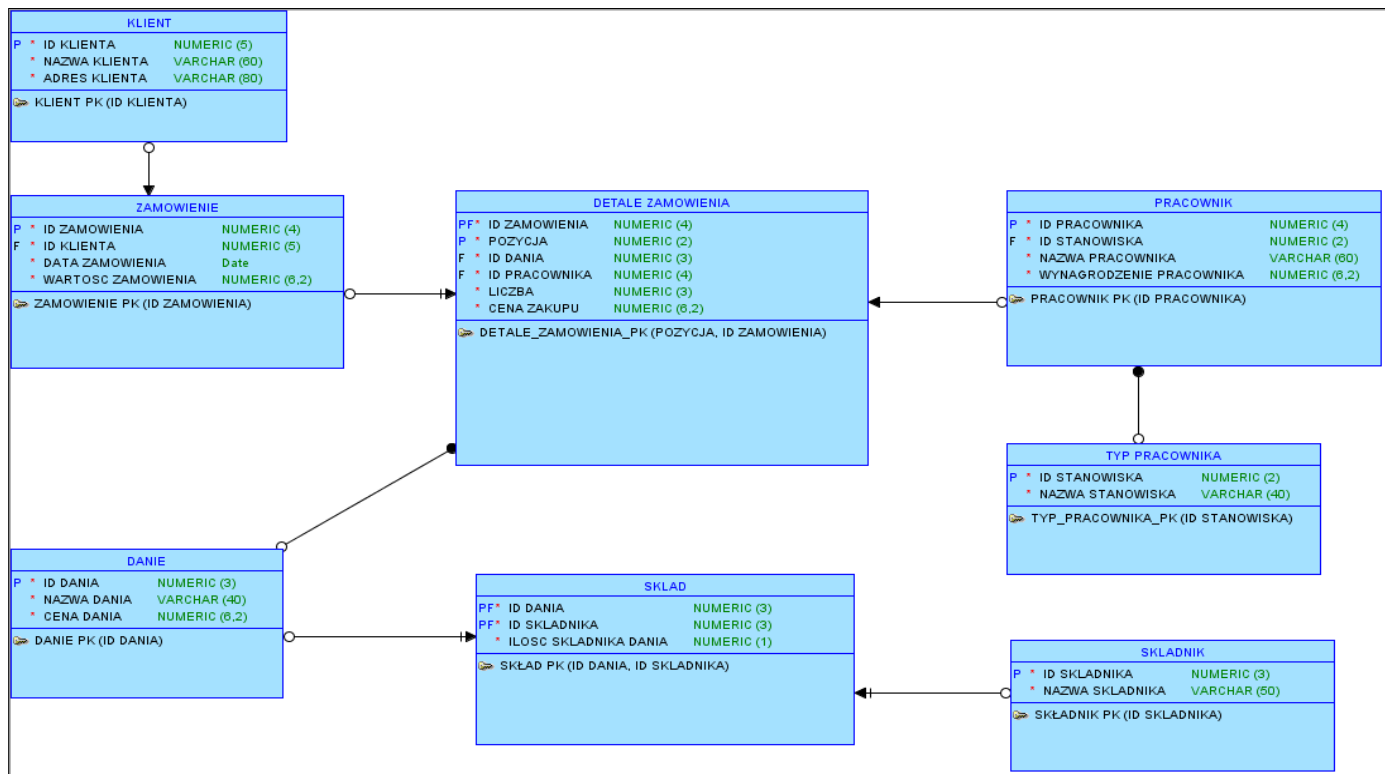
1. Analiza biznesowa

Projekt przedstawia bazę danych pizzerii. Lokal posiada w swojej ofercie kilka dań składających się ze składów. Każdy skład może posiadać wiele składników a każdy składnik ma swoją nazwę. Pizzeria posiada klientów, każdy z nich posiada adres który jest adresem dostawy. Każdy z klientów może dokonywać wiele zamówień. Baza samodzielnie generuje zamówienia w oparciu o model fakturowania wielopozycyjnego.

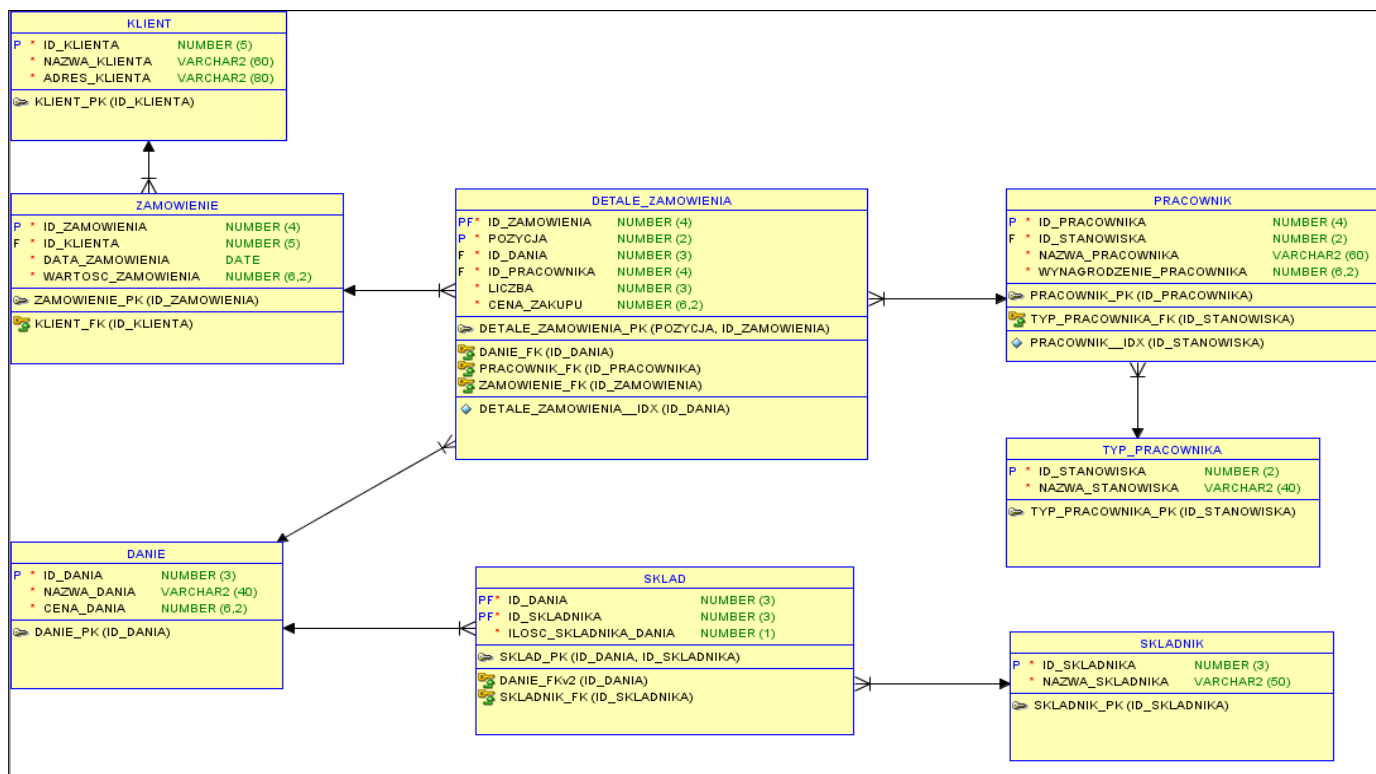
2. Model logiczny i relacyjny bazy

Model logiczny oraz relacyjny wykonany w programie Oracle SQL Developer Data Modeler dołączony jest do dokumentacji w folderze „oracledatamodeler_model_bazy” pod nazwą „model_logiczny_pizzeria.dmd”.

Model logiczny bazy wykonany w programie Oracle SQL Developer Data Modeler



Model relacyjny bazy wykonany w programie Oracle SQL Developer Data Modeler



Opis tabel oraz kolumn:

- KLIENT** – tabela przechowuje informacje na temat klientów. Składa się z kolumn:
 - **ID_KLIENTA (klucz główny)** – przechowuje identyfikator klienta
 - **NAZWA_KLIENTA** – przechowuje imię i nazwisko klienta
 - **ADRES_KLIENTA** – przechowuje pełny adres (ulica, miasto, kod pocztowy) klienta
- ZAMOWIENIE** – tabela będąca elementem modelu fakturowania wielopozycyjnego, przechowuje podstawowe informacje na temat zamówienia
 - **ID_ZAMOWIENIA (klucz główny)** – przechowuje identyfikator zamówienia
 - **ID_KLIENTA (klucz obcy)** – przechowuje identyfikator klienta
 - **DATA_ZAMOWIENIA** – przechowuje datę zamówienia
 - **WARTOSC_ZAMOWIENIA** – przechowuje wartość całego zamówienia, sumę cen wszystkich pozycji zamówienia
- DETALE_ZAMOWIENIA** – tabela będąca elementem modelu fakturowania wielopozycyjnego, przechowuje szczegóły każdej pozycji konkretnego zamówienia
 - **ID_ZAMOWIENIA (klucz główny, obcy)** – przechowuje identyfikator zamówienia
 - **POZYCJA (klucz główny)** – przechowuje numer pozycji zamówienia
 - **ID_DANCIA (klucz obcy)** – przechowuje identyfikator dania
 - **ID_PRACOWNIKA (klucz obcy)** – przechowuje identyfikator pracownika
 - **LICZBA** – określa ilość zakupionych dań w pozycji zamówienia
 - **CENA_ZAKUPU** – przechowuje iloczyn ceny dania i ilości zakupionych tych dań w jednej pozycji

- **DANIE** – tabela przechowuje podstawowe informacje na temat dań podawanych w lokalu
 - **ID_DANIA (klucz główny)** – przechowuje identyfikator dania
 - **NAZWA_DANIA** – przechowuje nazwę dania
 - **CENA_DANIA** – przechowuje cenę dania
- **SKŁAD** – tabela przechowująca składniki poszczególnych dań
 - **ID_DANIA (klucz główny, obcy)** – przechowuje identyfikator dania
 - **ID_SKŁADNIKA (klucz główny, obcy)** – przechowuje identyfikator składnika dania
 - **ILOSC_SKŁADNIKA_DANIA** – przechowuje ilość składników w danym daniu
- **SKŁADNIK** – tabela przechowująca składniki zawierające się w poszczególnych składach dań
 - **ID_SKŁADNIKA (klucz główny)** - przechowuje identyfikator składnika
 - **NAZWA_SKŁADNIKA** – przechowuje nazwę składnika
- **PRACOWNIK** – tabela przechowująca informacje na temat pracowników zatrudnionych w lokalu
 - **ID_PRACOWNIKA(klucz główny)** – przechowuje identyfikator pracownika
 - **ID_STANOWISKA(klucz obcy)** – przechowuje identyfikator stanowiska danego pracownika
 - **NAZWA_PRACOWNIKA** – przechowuje imię i nazwisko pracownika
 - **WYNAGRODZENIE_PRACOWNIKA** – przechowuje wartość wynagrodzenia pracownika
- **TYP_PRACOWNIKA** – tabela słownikowa przechowująca stanowiska na których zatrudnieni są pracownicy
 - **ID_STANOWISKA(klucz główny)** – przechowuje identyfikator stanowiska
 - **NAZWA_STANOWISKA** – przechowuje nazwę stanowiska

3. Oprogramowanie wdrożeniowe/usuwające bazę na ORACLE, generator zamówień:

- Skrypt wdrożeniowy znajduje się w folderze „oracle/tworzenie_bazy” pod nazwą „tworzenie_bazy.ddl” dołączonym wraz z dokumentacją
- Skrypt deinstalujący bazę znajduje się w folderze „oracle/tworzenie_bazy” pod nazwą „usuniecie_bazy.ddl” dołączonym wraz z dokumentacją.

Instalacja bazy danych na Oracle:

```
DROP TABLE danie CASCADE CONSTRAINTS;

DROP TABLE detale_zamowienia CASCADE CONSTRAINTS;

DROP TABLE klient CASCADE CONSTRAINTS;

DROP TABLE pracownik CASCADE CONSTRAINTS;

DROP TABLE sklad CASCADE CONSTRAINTS;

DROP TABLE skladnik CASCADE CONSTRAINTS;

DROP TABLE typ_pracownika CASCADE CONSTRAINTS;

DROP TABLE zamowienie CASCADE CONSTRAINTS;

CREATE TABLE danie (
  id_dania    NUMBER(3) NOT NULL,
  nazwa_dania VARCHAR2(40) NOT NULL,
  cena_dania  NUMBER(6,2) NOT NULL
);

ALTER TABLE danie ADD CONSTRAINT danie_pk PRIMARY KEY ( id_dania );

CREATE TABLE detale_zamowienia (
  id_zamowienia  NUMBER(4) NOT NULL,
  pozycja        NUMBER(2) NOT NULL,
  id_dania       NUMBER(3) NOT NULL,
  id_pracownika  NUMBER(4) NOT NULL,
  liczba         NUMBER(3) NOT NULL,
  cena_zakupu    NUMBER(6,2) NOT NULL
);

CREATE INDEX detale_zamowienia_idx ON
  detale_zamowienia ( id_dania ASC );

ALTER TABLE detale_zamowienia ADD CONSTRAINT detale_zamowienia_pk PRIMARY KEY (
  pozycja,id_zamowienia );
```

```

CREATE TABLE klient (
    id_klienta    NUMBER(5) NOT NULL,
    nazwa_klienta VARCHAR2(60) NOT NULL,
    adres_klienta VARCHAR2(80) NOT NULL
);

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

CREATE TABLE pracownik (
    id_pracownika      NUMBER(4) NOT NULL,
    id_stanowiska       NUMBER(2) NOT NULL,
    nazwa_pracownika    VARCHAR2(60) NOT NULL,
    wynagrodzenie_pracownika NUMBER(6,2) NOT NULL
);

CREATE INDEX pracownik__idx ON
    pracownik ( id_stanowiska ASC );

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

CREATE TABLE sklad (
    id_dania        NUMBER(3) NOT NULL,
    id_skladnika     NUMBER(3) NOT NULL,
    ilosc_skladnika_dania NUMBER(1) NOT NULL
);

ALTER TABLE sklad ADD CONSTRAINT sklad_pk PRIMARY KEY ( id_dania,id_skladnika );

CREATE TABLE skladnik (
    id_skladnika    NUMBER(3) NOT NULL,
    nazwa_skladnika VARCHAR2(50) NOT NULL
);

ALTER TABLE skladnik ADD CONSTRAINT skladnik_pk PRIMARY KEY ( id_skladnika );

CREATE TABLE typ_pracownika (
    id_stanowiska    NUMBER(2) NOT NULL,
    nazwa_stanowiska VARCHAR2(40) NOT NULL
);

ALTER TABLE typ_pracownika ADD CONSTRAINT typ_pracownika_pk PRIMARY KEY ( id_stanowiska );

CREATE TABLE zamowienie (
    id_zamowienia    NUMBER(4) NOT NULL,
    id_klienta        NUMBER(5) NOT NULL,
    data_zamowienia   DATE NOT NULL,
    wartosc_zamowienia NUMBER(6,2) NOT NULL
);

```

Relacje:

```
ALTER TABLE zamowienie ADD CONSTRAINT zamowienie_pk PRIMARY KEY ( id_zamowienia );

ALTER TABLE detale_zamowienia ADD CONSTRAINT danie_fk FOREIGN KEY ( id_dania )
REFERENCES danie ( id_dania );

ALTER TABLE sklad ADD CONSTRAINT danie_fkv2 FOREIGN KEY ( id_dania )
REFERENCES danie ( id_dania );

ALTER TABLE zamowienie ADD CONSTRAINT klient_fk FOREIGN KEY ( id_klienta )
REFERENCES klient ( id_klienta );

ALTER TABLE detale_zamowienia ADD CONSTRAINT pracownik_fk FOREIGN KEY ( id_pracownika )
REFERENCES pracownik ( id_pracownika );

ALTER TABLE sklad ADD CONSTRAINT skladnik_fk FOREIGN KEY ( id_skladnika )
REFERENCES skladnik ( id_skladnika );

ALTER TABLE pracownik ADD CONSTRAINT typ_pracownika_fk FOREIGN KEY ( id_stanowiska )
REFERENCES typ_pracownika ( id_stanowiska );

ALTER TABLE detale_zamowienia ADD CONSTRAINT zamowienie_fk FOREIGN KEY ( id_zamowienia )
REFERENCES zamowienie ( id_zamowienia );
```

Uzupełnianie tabel danych stałych:

Ten fragment skryptu został wygenerowany na podstawie plików .csv za pomocą narzędzia „Import Data” w programie Oracle SQL Developer. Pliki .csv zostały dołączone do dokumentacji w folderze „oracle”. Tabele „ZAMOWIENIE” oraz „DETALE_ZAMOWIENIA” uzupełniane są z wykorzystaniem generatora zamówień, którego poszczególne funkcje opisane są w dalszej części dokumentacji.

```
INSERT INTO TYP_PRACOWNIKA (ID_STANOWISKA, NAZWA_STANOWISKA)
VALUES (1, 'Kucharz');
INSERT INTO TYP_PRACOWNIKA (ID_STANOWISKA, NAZWA_STANOWISKA)
VALUES (2, 'Kelner');

INSERT INTO PRACOWNIK (ID_PRACOWNIKA, ID_STANOWISKA, NAZWA_PRACOWNIKA,
WYNAGRODZENIE_PRACOWNIKA)
VALUES (1, 1, 'Maciej Barylka', 2500.00);
INSERT INTO PRACOWNIK (ID_PRACOWNIKA, ID_STANOWISKA, NAZWA_PRACOWNIKA,
WYNAGRODZENIE_PRACOWNIKA)
VALUES (2, 1, 'Krzysztof Morawski', 1950.00);
INSERT INTO PRACOWNIK (ID_PRACOWNIKA, ID_STANOWISKA, NAZWA_PRACOWNIKA,
WYNAGRODZENIE_PRACOWNIKA)
VALUES (3, 2, 'Alojzy Opania', 1700.00);
INSERT INTO PRACOWNIK (ID_PRACOWNIKA, ID_STANOWISKA, NAZWA_PRACOWNIKA,
WYNAGRODZENIE_PRACOWNIKA)
VALUES (4, 2, 'Marcin Korytko', 1700.00);
```

```

INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (1.0, 'Mozzarella');
INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (2.0, 'Parmezan');
INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (3.0, 'Cheddar');
INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (4.0, 'Ser Plesniowy');
INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (5.0, 'Szynka');
INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (6.0, 'Pieczarki');
INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (7.0, 'Ananas');
INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (8.0, 'Sos Pomidorowy');
INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (9.0, 'Makaron');
INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (10.0, 'Mieso Wolowe');
INSERT INTO skladnik (ID_SKLADNIKA, NAZWA_SKLADNIKA)
VALUES (11.0, 'Boczek');

```

```

INSERT INTO DANIE (ID_DANIA, NAZWA_DANIA, CENA_DANIA)
VALUES (1.0, 'Pizza Margeritha', 14.5);
INSERT INTO DANIE (ID_DANIA, NAZWA_DANIA, CENA_DANIA)
VALUES (2.0, 'Pizza Prosciutto', 17.8);
INSERT INTO DANIE (ID_DANIA, NAZWA_DANIA, CENA_DANIA)
VALUES (3.0, 'Pizza Hawajska', 18.0);
INSERT INTO DANIE (ID_DANIA, NAZWA_DANIA, CENA_DANIA)
VALUES (4.0, 'Pizza Serowa', 16.5);
INSERT INTO DANIE (ID_DANIA, NAZWA_DANIA, CENA_DANIA)
VALUES (5.0, 'Spaghetti Bolognese', 15.99);
INSERT INTO DANIE (ID_DANIA, NAZWA_DANIA, CENA_DANIA)
VALUES (6.0, 'Spaghetti Carbonara', 16.99);

```

```

INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (1.0,1.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (1.0,8.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (2.0,1.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (2.0,5.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (2.0,6.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (2.0,8.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (3.0,1.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (3.0,5.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (3.0,7.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (3.0,8.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (4.0,1.0,2.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (4.0,2.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (4.0,3.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (4.0,4.0,1.0);

```

```

INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (4.0,8.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (5.0,8.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (5.0,9.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (5.0,10.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (6.0,2.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (6.0,9.0,1.0);
INSERT INTO sklad (ID_DANIA, ID_SKLADNIKA, ILOSC_SKLADNIKA_DANIA) VALUES (6.0,11.0,1.0);

```

```

INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (1.0, 'Adrian Ciok', 'Grabianowska 5 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (2.0, 'Jakub Ciok', 'Teodorów 5 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (3.0, 'Maria Górską', 'Warszawska 80 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (4.0, 'Bogdan Górski', 'Warszawska 80 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (5.0, 'Natalia Prontek', 'Piłsudskiego 56/90 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (6.0, 'Marek Lapiński', 'Platynowa 5/30 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (7.0, 'Alicja Wyrzykowska', 'Moniuszki 30 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (8.0, 'Adrian Łaski', 'Niepodległości 10/3 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (9.0, 'Katarzyna Pasikonik', 'Wyszyńskiego 8/13 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (10.0, 'Kamil Boberek', 'Ogińskich 13 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (11.0, 'Marek Słowak', 'Żelków-Kolonia 13 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (12.0, 'Leszek Chmielewski', 'Unitów Podlaskich 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (13.0, 'Stefan Górzyński', 'Kropelkowa 70/3 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (14.0, 'Andrzej Wróblewski', 'Sw Krolowej Jadwigi 22a 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (15.0, 'Dawid Tokarzewski', 'Uroczysko 13 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (16.0, 'Piotr Pogorzelski', 'Katedralna 90/90 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (17.0, 'Karolina Smolińska', 'Powstańców Śląskich 87/100 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (18.0, 'Dagmara Góra-Sawicka', 'Unitów Podlaskich 23/3 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (19.0, 'Joanna Pałasz', 'Romanówka 1/32 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (20.0, 'Maciej Turkowski', 'Maczka 13/3 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (21.0, 'Jagoda Debowska', 'Konarskiego 12 08-110 Siedlce');
INSERT INTO klient (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (22.0, 'Michał Jaworski', 'Pomorska 3/23 08-110 Siedlce');

```



```

INSERT INTO KLIENT (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (23.0, 'Arkadiusz Jaworski', 'Pomorska 3/23 08-110 Siedlce');
INSERT INTO KLIENT (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (24.0, 'Tomasz Chada', 'Daszynskiego 20/32 08-110 Siedlce');
INSERT INTO KLIENT (ID_KLIENTA, NAZWA_KLIENTA, ADRES_KLIENTA)
VALUES (25.0, 'Kinga Ksiezna', 'Weterynaryjna 12 08-110 Siedlce');

```

Funkcje:

Funkcja losująca liczbę całkowitą z przedziału od 1 do zadanej wartości maksymalnej:

```

CREATE OR REPLACE FUNCTION FN_LOSUJ(V_MAX NUMBER)
RETURN NUMBER AS
  V_WYLOSOWANA NUMBER;
BEGIN
  V_WYLOSOWANA := ROUND(DBMS_RANDOM.VALUE(1,v_max));
RETURN V_WYLOSOWANA;
END FN_LOSUJ;

```

Funkcja losująca identyfikator dania(„ID_DANIA”) na potrzeby generatora zamówień. Przypisuje ona na zmienne maksymalny oraz minimalny identyfikator dania a następnie za pomocą funkcji „DBMS_RANDOM” losuje identyfikator:

```

create or replace FUNCTION FN_LOSUJ_DANIE
RETURN NUMBER AS V_WYLOSOWANA NUMBER;
V_MIN NUMBER;
V_MAX NUMBER;
BEGIN
  SELECT MIN(DANIE.ID_DANIA) INTO V_MIN FROM DANIE;
  SELECT MAX(DANIE.ID_DANIA) INTO V_MAX FROM DANIE;
  V_WYLOSOWANA := ROUND(DBMS_RANDOM.VALUE(V_MIN,V_MAX));
RETURN V_WYLOSOWANA;
END FN_LOSUJ_DANIE;

```

Funkcja losująca identyfikator klienta („ID_KLIENTA”) na potrzeby generatora zamówień. Przypisuje ona na zmienne maksymalny oraz minimalny identyfikator klienta a następnie za pomocą funkcji „DBMS_RANDOM” losuje identyfikator:

```

create or replace FUNCTION FN_LOSUJ_KLIENTA
RETURN NUMBER AS V_WYLOSOWANY_KLIENT NUMBER;
V_MIN NUMBER;
V_MAX NUMBER;
BEGIN
  SELECT MIN(ID_KLIENTA) INTO V_MIN FROM KLIENT;
  SELECT MAX(ID_KLIENTA) INTO V_MAX FROM KLIENT;
  V_WYLOSOWANY_KLIENT := ROUND(DBMS_RANDOM.VALUE(V_MIN,V_MAX));
RETURN V_WYLOSOWANY_KLIENT;
END FN_LOSUJ_KLIENTA;

```

Funkcja sumująca ceny zakupu poszczególnych pozycji o zadanym identyfikatorze zamówienia („ID_ZAMOWIENIA”) na potrzeby generatora zamówień. Wykorzystuje ona funkcję „SUM” służącą do obliczania sumy wartości:

```
CREATE OR REPLACE FUNCTION FN_SUMA
(
  V_ID_ZAMOWIENIA IN NUMBER
) RETURN NUMBER AS V_WARTOSC_ZAMOWIENIA NUMBER;
BEGIN
  SELECT SUM(CENA_ZAKUPU) INTO V_WARTOSC_ZAMOWIENIA FROM DETALE_ZAMOWIENIA
    WHERE ID_ZAMOWIENIA = V_ID_ZAMOWIENIA;
  RETURN V_WARTOSC_ZAMOWIENIA;
END FN_SUMA;
```

Funkcja losująca identyfikator kucharza („ID_KUCHARZA”) na potrzeby generatora zamówień. Przypisuje ona na zmienne wartości maksymalne oraz minimalne identyfikatorów dla pracowników których identyfikator stanowiska („ID_STANOWISKA”) jest równy 1. Identyfikator ten jest identyfikatorem kucharza.

```
create or replace FUNCTION FN_LOSUJ_KUCHARZA
RETURN NUMBER AS V_ID_KUCHARZA NUMBER;
V_MIN NUMBER;
V_MAX NUMBER;
BEGIN
  SELECT MIN(ID_PRACOWNIKA) INTO V_MIN FROM PRACOWNIK WHERE ID_STANOWISKA = 1;
  SELECT MAX(ID_PRACOWNIKA) INTO V_MAX FROM PRACOWNIK WHERE ID_STANOWISKA = 1;
  V_ID_KUCHARZA := ROUND(DBMS_RANDOM.VALUE(V_MIN,V_MAX));
  RETURN V_ID_KUCHARZA;
END FN_LOSUJ_KUCHARZA;
```

Sekwencja

Sekwencja ta odpowiedzialna jest za nadawanie kolejno identyfikatorów zamówienia („ID_ZAMOWIENIA”) każdemu zamówieniu generowanemu przez generator zamówień.

```
CREATE SEQUENCE SEQ_ID_ZAMOWIENIA
MINVALUE 1
START WITH 1
INCREMENT BY 1;
```

Procedura – generator zamówień

Procedura ta uruchamia odpowiednie zaimplementowane funkcje opisane wyżej oraz sekwencję a następnie na podstawie wyników ich działań dodaje odpowiednio wartości do kolumn w tabelach „ZAMOWIENIE” i „DETALE_ZAMOWIENIE” tworząc cały wpis zamówienia oraz odświeża perspektywy zmaterializowane.

```
create or replace PROCEDURE PR_GENERUJ_ZAMOWIENIA
AS
ID_ZAM NUMBER;
ILOSC_POZYCJI NUMBER;
POZYCJA NUMBER;
V_ID_DANIA NUMBER;
V_ID_KLIENTA NUMBER;
ID_KUCHARZA NUMBER;
LICZBA NUMBER;
CENA NUMBER;
BEGIN
FOR a in 1..3 LOOP
  ID_ZAM := SEQ_ID_ZAMOWIENIA.nextval;
  ILOSC_POZYCJI := FN_LOSUJ(4);
  POZYCJA := 1;
  V_ID_KLIENTA := FN_LOSUJ_KLIENTA();
  INSERT INTO ZAMOWIENIE
    VALUES(ID_ZAM,V_ID_KLIENTA,CURRENT_DATE,0);
  FOR b IN 1..ILOSC_POZYCJI LOOP
    V_ID_DANIA := FN_LOSUJ_DANIE();
    LICZBA := FN_LOSUJ(3);
    SELECT CENA_DANIA INTO CENA FROM DANIE WHERE ID_DANIA = V_ID_DANIA;
    INSERT INTO DETALE_ZAMOWIENIA
      VALUES(ID_ZAM,POZYCJA,V_ID_DANIA,FN_LOSUJ_KUCHARZA(),LICZBA,LICZBA*CENA);
    POZYCJA := POZYCJA +1;
  END LOOP;
  UPDATE ZAMOWIENIE SET WARTOSC_ZAMOWIENIA = FN_SUMA(ID_ZAM) WHERE
ID_ZAMOWIENIA = ID_ZAM;
END LOOP;
DBMS_MVIEW.REFRESH('ZYSK_STYCZNIOWY');
DBMS_MVIEW.REFRESH('ZYSK_STYCZNIOWY_SZCZEGOLY');
DBMS_MVIEW.REFRESH('KTO_PRZYGOTOWAL_DANIE');
END PR_GENERUJ_ZAMOWIENIA;
```

Wyzwalacz

Mechanizm ten usuwa wiersze w tabeli „DETALE_ZAMOWIENIA” których identyfikator zamówienia („ID_ZAMOWIENIA”) jest identyczne z identyfikatorem zamówienia usuwanego wiersza z tabeli „ZAMOWIENIE”. Ma on na celu „posprzątanie” w tabeli „DETALE_ZAMOWIENIA” zanim zostanie usunięte zamówienie w tabeli „ZAMOWIENIE”.

```
CREATE OR REPLACE TRIGGER DEL_ZAMOWIENIE
BEFORE DELETE ON ZAMOWIENIE
FOR EACH ROW
BEGIN
  DELETE FROM DETALE_ZAMOWIENIA
  WHERE ID_ZAMOWIENIA = :OLD.ID_ZAMOWIENIA;
END;
```

Perspektywy oraz perspektywy zmaterializowane

- Perspektywa zastępuje wielokrotne wykonywanie zapytania do bazy danych. Jest to struktura dynamiczna która wykonywana jest w momencie odwołania się do niej.
- Perspektywa zmaterializowana w odróżnieniu od zwykłej perspektywy przechowuje dane i wykorzystuje fizyczną przestrzeń dyskową bazy danych. Po każdym dodaniu lub usunięciu wierszy do tabel dla których napisaliśmy taką perspektywę należy ją zaktualizować za pomocą funkcji „DBMS_MVIEW.REFRESH”. Stosowane są głównie w celu poprawy wydajności dla zapytań.

Perspektywy te wyświetlają typowe wskaźniki biznesowe – zysk z dzisiejszego dnia oraz zysk z konkretnego miesiąca (Stycznia), podzielone są na wersje ogólną zawierającą tylko kwotę i ilość sprzedanych dań oraz szczegółową zawierającą wykaz zysków z konkretnych dań. Kolejna perspektywa wyświetla dane klientów oraz kwoty wydane przez nich na zamówienia. Dodatkowo istnieje perspektywa pokazująca który pracownik (kucharz) przygotował jakie danie. Ostatnia z wymienionych perspektyw przygotowana jest z myślą o właścicielu firmy i ewentualnych skargach na serwowane posiłki.

```
CREATE MATERIALIZED VIEW ZYSK_STYCZNIOWY AS
SELECT SUM(CENA_ZAKUPU) AS "ZYSK W STYCZNIU", SUM(LICZBA) AS "ILOSC SPRZEDANYCH DAŃ"
FROM ZAMOWIENIE, DETALE_ZAMOWIENIA
WHERE ZAMOWIENIE.ID_ZAMOWIENIA = DETALE_ZAMOWIENIA.ID_ZAMOWIENIA
AND EXTRACT(MONTH FROM DATA_ZAMOWIENIA) = 1;

CREATE MATERIALIZED VIEW ZYSK_STYCZNIOWY_SZCZEGOLY AS
SELECT DISTINCT DETALE_ZAMOWIENIA.ID_DANIA AS "NUMER DANIA", NAZWA_DANIA AS "NAZWA
DANIA", SUM(CENA_ZAKUPU) AS "SUMA ZYSKU", SUM(LICZBA) AS "ILOSC SPRZEDANYCH DAN"
FROM ZAMOWIENIE, DETALE_ZAMOWIENIA, DANIE
WHERE DETALE_ZAMOWIENIA.ID_ZAMOWIENIA = ZAMOWIENIE.ID_ZAMOWIENIA
AND DETALE_ZAMOWIENIA.ID_DANIA = DANIE.ID_DANIA
AND EXTRACT(MONTH FROM DATA_ZAMOWIENIA) = 1
GROUP BY DETALE_ZAMOWIENIA.ID_DANIA, NAZWA_DANIA
```

```

ORDER BY DETALE_ZAMOWIENIA.ID_DANIA ASC;

CREATE MATERIALIZED VIEW KTO_PRZYGOTOWAL_DANIE AS
SELECT NAZWA_PRACOWNIKA AS "IMIE I NAZWISKO KUCHARZA", ID_ZAMOWIENIA AS "NUMER
ZAMOWIENIA", POZYCJA AS "POZYCJA ZAMOWIENIA", NAZWA_DANIA AS "NAZWA DANIA" FROM
PRACOWNIK,DETALE_ZAMOWIENIA, DANIE
WHERE DETALE_ZAMOWIENIA.ID_PRACOWNIKA = PRACOWNIK.ID_PRACOWNIKA AND
DETALE_ZAMOWIENIA.ID_DANIA = DANIE.ID_DANIA
ORDER BY ID_ZAMOWIENIA ASC;

CREATE OR REPLACE VIEW ZYSK_DZISIEJSZY AS
SELECT CURRENT_DATE AS "DATA", SUM(CENA_ZAKUPU) AS "ZYSK DZISIEJSZY", SUM(LICZBA) AS
"ILOSC SPRZEDANYCH DAŃ"
FROM ZAMOWIENIE,DETALE_ZAMOWIENIA
WHERE ZAMOWIENIE.ID_ZAMOWIENIA = DETALE_ZAMOWIENIA.ID_ZAMOWIENIA
AND TRUNC(DATA_ZAMOWIENIA) = TRUNC(CURRENT_DATE);

CREATE OR REPLACE VIEW ZYSK_DZISIEJSZY_SZCZEGOLY AS
SELECT DISTINCT CURRENT_DATE AS "DATA", DETALE_ZAMOWIENIA.ID_DANIA AS "NUMER DANIA",
NAZWA_DANIA AS "NAZWA DANIA", SUM(CENA_ZAKUPU) AS "SUMA ZYSKU", SUM(LICZBA) AS
"ILOSC SPRZEDANYCH DAN"
FROM ZAMOWIENIE, DETALE_ZAMOWIENIA, DANIE
WHERE DETALE_ZAMOWIENIA.ID_ZAMOWIENIA = ZAMOWIENIE.ID_ZAMOWIENIA
AND DETALE_ZAMOWIENIA.ID_DANIA = DANIE.ID_DANIA
AND TRUNC(DATA_ZAMOWIENIA) = TRUNC(CURRENT_DATE)
GROUP BY DETALE_ZAMOWIENIA.ID_DANIA, NAZWA_DANIA
ORDER BY DETALE_ZAMOWIENIA.ID_DANIA ASC;

CREATE OR REPLACE VIEW ZAM_KLIENCI_WART_DZISIAJ AS
SELECT CURRENT_DATE as "DATA", ID_ZAMOWIENIA AS "NUMER ZAMOWIENIA", NAZWA_KLIENTA
AS "IMIE I NAZWISKO KLIENTA", ADRES_KLIENTA AS "ADRES KLIENTA", WARTOSC_ZAMOWIENIA AS
"WARTOSC ZAMOWIENIA"
FROM ZAMOWIENIE, KLIENT
WHERE ZAMOWIENIE.ID_KLIENTA = KLIENT.ID_KLIENTA
AND TRUNC(DATA_ZAMOWIENIA) = TRUNC(CURRENT_DATE)
ORDER BY ID_ZAMOWIENIA ASC;

```

Obsługa zdarzeń, cykliczne działanie generatora zamówień

Za cykliczne uruchamianie procedury generowania zamówień odpowiedzialny jest „job” uruchamiający procedurę generowania zamówień automatycznie co 6 godzin. Uruchamiany jest podczas ładowania bazy danych na serwer Oracle.

W celu wyłączenia go należy wpisać i wykonać następującą linijkę kodu:

dbms_scheduler.DISABLE('GENERATOR_ZAMOWIEN');

```

BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
    job_name      => 'GENERATOR_ZAMOWIEN',
    job_type      => 'STORED_PROCEDURE',

```

```
job_action      => 'PR_GENERUJ_ZAMOWIENIA',
start_date      => SYSDATE,
repeat_interval => 'FREQ=HOURLY;INTERVAL=6',
auto_drop       => FALSE,
comments        => 'GENERATOR ZAMOWIEN CO 6H';
dbms_scheduler.ENABLE('GENERATOR_ZAMOWIEN');
END;
```

Deinstalacja bazy na Oracle

Skrypt deinstalujący bazę danych usuwa tabele wraz z ich zawartością, funkcje, procedurę generatora, sekwencję, perspektywy oraz cykliczne działanie generatora. Wyzwalacz usuwany jest automatycznie wraz z usunięciem tabeli dla której został napisany.

```
DROP TABLE danie CASCADE CONSTRAINTS;
DROP TABLE detale_zamowienia CASCADE CONSTRAINTS;
DROP TABLE klient CASCADE CONSTRAINTS;
DROP TABLE pracownik CASCADE CONSTRAINTS;
DROP TABLE sklad CASCADE CONSTRAINTS;
DROP TABLE skladnik CASCADE CONSTRAINTS;
DROP TABLE typ_pracownika CASCADE CONSTRAINTS;
DROP TABLE zamowienie CASCADE CONSTRAINTS;

DROP FUNCTION FN_LOSUJ;
DROP FUNCTION FN_LOSUJ_DANIE;
DROP FUNCTION FN_LOSUJ_KLIENTA;
DROP FUNCTION FN_LOSUJ_KUCHARZA;
DROP FUNCTION FN_SUMA;
DROP PROCEDURE PR_GENERUJ_ZAMOWIENIA;
DROP SEQUENCE SEQ_ID_ZAMOWIENIA;
DROP MATERIALIZED VIEW ZYSK_STYCZNIOWY;
DROP MATERIALIZED VIEW ZYSK_STYCZNIOWY_SZCZEGOLY;
DROP MATERIALIZED VIEW KTO_PRZYGOTOWAL_DANIE;
DROP VIEW ZYSK_DZISIEJSZY;
DROP VIEW ZYSK_DZISIEJSZY_SZCZEGOLY;
DROP VIEW ZAM_KLIENCI_WART_DZISIAJ;
DBMS_SCHEDULER.DROP_JOB('GENERATOR_ZAMOWIEN');
```

4. Instalacja/deinstalacja bazy danych, sprawdzenie poprawności instalacji

- Aby zainstalować bazę w środowisku Oracle należy uruchomić program Oracle SQL Developer, zalogować się na serwer a następnie wprowadzić skrypt „tworzenie_bazy.ddl” znajdujący się w folderze „oracle” i uruchomić wykonywanie poleceń SQL.
Aby odinstalować bazę w środowisku Oracle należy uruchomić program Oracle SQL Developer, zalogować się na serwer, wprowadzić skrypt „usuniecie_bazy.ddl” znajdujący się w folderze „oracle” i uruchomić wykonywanie poleceń SQL.

Sprawdzenie poprawności instalacji bazy

Aby sprawdzić poprawność instalacji bazy należy odpytać serwer poniższymi zapytaniami i porównać zawartość tabel ze zrzutami ekranu dołączonymi do dokumentacji:

- Select * from DANIE;

	ID_DANIA	NAZWA_DANIA	CENA_DANIA
1	1	Pizza Margeritha	14,5
2	2	Pizza Prosciutto	17,8
3	3	Pizza Hawajska	18
4	4	Pizza Serowa	16,5
5	5	Spaghetti Bolognese	15,99
6	6	Spaghetti Carbonara	16,99

- Select * from TYP_PRACOWNIKA;

	ID_STANOWISKA	NAZWA_STANOWISKA
1	1	Kucharz
2	2	Kelner

- Select * from PRACOWNIK;

	ID_PRACOWNIKA	ID_STANOWISKA	NAZWA_PRACOWNIKA	WYNAGRODZENIE_PRACOWNIKA
1	1	1	Maciej Barylka	2500
2	2	2	Krzysztof Morawski	1950
3	3	3	Alojzy Opania	1700
4	4	4	Marcin Korytko	1700

- Select * from SKLADNIK;

ID_SKLADNIKA	NAZWA_SKLADNIKA
1	1 Mozzarella
2	2 Parmezan
3	3 Cheddar
4	4 Ser Plesniowy
5	5 Szynka
6	6 Pieczarki
7	7 Ananas
8	8 Sos Pomidorowy
9	9 Makaron
10	10 Mieso Wolowe
11	11 Boczek

- Select * from KLIENT;

ID_KLIENTA	NAZWA_KLIENTA	ADRES_KLIENTA
1	1 Adrian Ciok	Grabianowska 5 08-110 Siedlce
2	2 Jakub Ciok	Teodorów 5 08-110 Siedlce
3	3 Maria Górską	Warszawska 80 08-110 Siedlce
4	4 Bogdan Górski	Warszawska 80 08-110 Siedlce
5	5 Natalia Prontek	Piłsudskiego 56/90 08-110 Siedlce
6	6 Marek Lapiński	Platynowa 5/30 08-110 Siedlce
7	7 Alicja Wyrzykowska	Moniuszki 30 08-110 Siedlce
8	8 Adrian Łaski	Niepodległości 10/3 08-110 Siedlce
9	9 Katarzyna Pasikonik	Wyszyńskiego 8/13 08-110 Siedlce
10	10 Kamil Boberek	Ogińskich 13 08-110 Siedlce
11	11 Marek Słowak	Żelków-Kolonia 13 08-110 Siedlce
12	12 Leszek Chmielewski	Unitów Podlaskich 08-110 Siedlce
13	13 Stefan Górzynski	Kropelkowa 70/3 08-110 Siedlce
14	14 Andrzej Wróblewski	Sw Krolowej Jadwigi 22a 08-110 Siedlce
15	15 Dawid Tokarzewski	Uroczysko 13 08-110 Siedlce
16	16 Piotr Pogorzelski	Katedralna 90/90 08-110 Siedlce
17	17 Karolina Smolińska	Powstańców Śląskich 87/100 08-110 Siedlce
18	18 Dagmara Góra-Sawicka	Unitów Podlaskich 23/3 08-110 Siedlce
19	19 Joanna Pałasz	Romanówka 1/32 08-110 Siedlce
20	20 Maciej Turkowski	Maczka 13/3 08-110 Siedlce
21	21 Jagoda Debowska	Konarskiego 12 08-110 Siedlce
22	22 Michał Jaworski	Pomorska 3/23 08-110 Siedlce
23	23 Arkadiusz Jaworski	Pomorska 3/23 08-110 Siedlce
24	24 Tomasz Chada	Daszyńskiego 20/32 08-110 Siedlce
25	25 Kinga Ksiezna	Weterynaryjna 12 08-110 Siedlce

- Select * from SKLAD;

ID_DANIA	ID_SKLADNIKA	ILOSC_SKLADNIKA_DANIA
1	1	1
2	1	8
3	2	1
4	2	5
5	2	6
6	2	8
7	3	1
8	3	5
9	3	7
10	3	8
11	4	1
12	4	2
13	4	3
14	4	4
15	4	8
16	5	8
17	5	9
18	5	10
19	6	2
20	6	9
21	6	11

5. Działanie perspektyw analitycznych – zrzuty ekranu z programu Oracle SQL Developer

- Perspektywa zmaterializowana „KTO_PRZYGOTOWAŁ_DANIE”

	IMIE I NAZWISKO KUCHARZA	NUMER ZAMOWIENIA	POZYCJA ZAMOWIENIA	NAZWA DANIA
1	Maciej Barylka	4	1	Pizza Hawajska
2	Krzysztof Morawski	4	2	Pizza Hawajska
3	Krzysztof Morawski	5	1	Pizza Hawajska
4	Maciej Barylka	1	2	Pizza Serowa
5	Krzysztof Morawski	2	2	Pizza Serowa
6	Krzysztof Morawski	1	1	Spaghetti Bolognese
7	Krzysztof Morawski	3	1	Spaghetti Bolognese
8	Krzysztof Morawski	2	1	Spaghetti Carbonara
9	Maciej Barylka	4	3	Spaghetti Carbonara

- Perspektywa „ZAM_KLIENCI_WART_DZISIAJ”

	DATA	NUMER ZAMOWIENIA	IMIE I NAZWISKO KLIENTA	ADRES KLIENTA	WARTOSC ZAMOWIENIA
1	17/01/22		Michał Jaworski	Pomorska 3/23 08-110 Siedlce	64,98
2	17/01/22		Maria Górka	Warszawska 80 08-110 Siedlce	66,49
3	17/01/22		Joanna Pałasz	Romanówka 1/32 08-110 Siedlce	15,99
4	17/01/22		Piotr Pogorzelski	Katedralna 90/90 08-110 Siedlce	105,98
5	17/01/22		Katarzyna Pasikonik	Wyszyńskiego 8/13 08-110 Siedlce	36

- Perspektywa zmaterializowana „ZYSK_STYCZNIOWY”

	ZYSK W STYCZNIU	ILOSC SPRZEDANYCH DAŃ
1	289,44	17

- Perspektywa zmaterializowana „ZYSK_STYCZNIOWY_SZCZEGOLY”

	NUMER DANIA	NAZWA DANIA	SUMA ZYSKU	ILOSC SPRZEDANYCH DAN
1	6	Spaghetti Carbonara	50,97	3
2	4	Pizza Serowa	82,5	5
3	3	Pizza Hawajska	108	6
4	5	Spaghetti Bolognese	47,97	3

- Perspektywa „ZYSK_DZISIEJSZY”

	DATA	ZYSK DZISIEJSZY	ILOSC SPRZEDANYCH DAŃ
1	17/01/22	289,44	17

- Perspektywa „ZYSK_DZISIEJSZY_SZCZEGOLY”

	DATA	NUMER DANIA	NAZWA DANIA	SUMA ZYSKU	ILOSC SPRZEDANYCH DAN
1	17/01/22	3	Pizza Hawajska	108	6
2	17/01/22	4	Pizza Serowa	82,5	5
3	17/01/22	5	Spaghetti Bolognese	47,97	3
4	17/01/22	6	Spaghetti Carbonara	50,97	3

6.1. Działanie perspektyw analitycznych – zrzuty ekranu raportów PDF

Pliki z przykładowymi raportami w formacie PDF utworzonymi w programie „JasperReports” oraz pliki programu „JasperReports” załączone są do dokumentacji w folderze „raporty”.

- Raport z programu „JasperReports” utworzony na podstawie perspektywy „ZYSK_STYCZNIOWY_SZCZEGÓŁOWY”

ZYSK SZCZEGÓŁOWY STYCZEŃ 2017			
poniedziałek 23 styczeń 2017			
NUMER DANIA	NAZWA DANIA	SUMA ZYSKU	ILOŚĆ SPRZEDANYCH DAŃ
1	Pizza Margeritha	362.5	25
2	Pizza Prosciutto	623	35
3	Pizza Hawajska	612	34
4	Pizza Serowa	429	26
5	Spaghetti Bolognese	415.74	26
6	Spaghetti Carbonara	254.85	15

- Raport z programu „JasperReports” utworzony na podstawie perspektywy „ZYSK_STYCZNIOWY”

ZYSK OGÓLNY STYCZEŃ 2017	
poniedziałek 23 styczeń 2017	
ZYSK W STYCZNIU	ILOŚĆ SPRZEDANYCH DAŃ
2697.09	161

- Raport z programu „JasperReports” utworzony na podstawie perspektywy „ZYSK_DZISIEJSZY_SZCZEGOLY”

<div>ZYSK SZCZEGÓŁOWY</div> <div>poniedziałek 23 styczeń 2017</div>			
NUMER DANIA	NAZWA DANIA	SUMA ZYSKU	IŁOŚĆ SPRZEDANYCH DAŃ
1	Pizza Margeritha	43.5	3
2	Pizza Prosciutto	53.4	3
3	Pizza Hawajska	180	10
4	Pizza Serowa	49.5	3
6	Spaghetti Carbonara	84.95	5

- Raport z programu „JasperReports” utworzony na podstawie perspektywy „ZYSK_DZISIEJSZY”

<div>ZYSK DZISIEJSZY OGÓLNY</div> <div>poniedziałek 23 styczeń 2017</div>	
ZYSK DZISIEJSZY	IŁOŚĆ SPRZEDANYCH DAŃ
411.35	24

- Raport z programu JasperReports utworzony na podstawie perspektywy „ZAM_KLIENCI_WART_DZISIAJ”

DZISIEJSZE ZAMÓWIENIA			
poniedziałek 23 styczeń 2017			
NUMER ZAMÓWIENIA	IMIĘ I NAZWISKO KLIENTA	ADRES KLIENTA	WARTOŚĆ ZAMÓWIENIA
26	Stefan Górzyński	Kropelkowa 70/3 08-110 Siedlce	85.5
27	Stefan Górzyński	Kropelkowa 70/3 08-110 Siedlce	77.48
28	Arkadiusz Jaworski	Pomorska 3/23 08-110 Siedlce	50.97
29	Maria Górską	Warszawska 80 08-110 Siedlce	108
30	Dawid Tokarzewski	Uroczysko 13 08-110 Siedlce	89.4

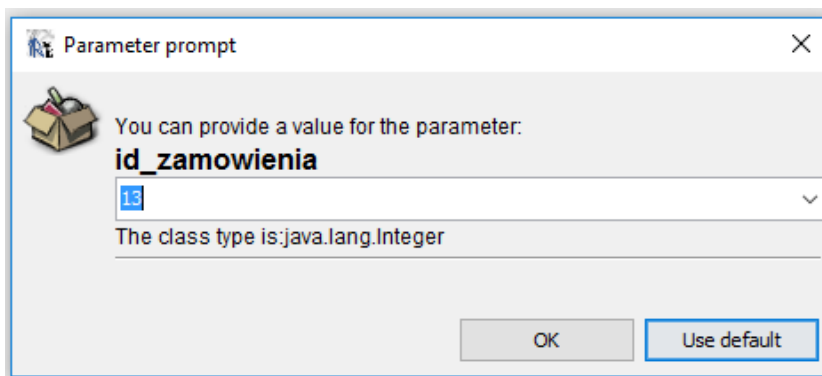
- Fragment raportu z programu JasperReports utworzonego na podstawie perspektywy „KTO_PRZYGOTOWAL_DANIE”

SPIS DAŃ I ZAMÓWIEŃ PRZYGOTOWANYCH PRZEZ DANEGO KUCHARZA			
niedziela 22 styczeń 2017			
IMIĘ I NAZWISKO	NUMER ZAMÓWIENIA	POZYCJA	NAZWA DANIA
Krzysztof Morawski	1	1	Pizza Serowa
Maciej Barylka	2	1	Spaghetti Carbonara
Maciej Barylka	2	2	Pizza Hawajska
Maciej Barylka	2	3	Pizza Prosciutto
Krzysztof Morawski	3	3	Pizza Margeritha
Maciej Barylka	3	2	Pizza Margeritha
Maciej Barylka	3	1	Pizza Margeritha
Maciej Barylka	4	2	Pizza Serowa
Maciej Barylka	4	3	Spaghetti Bolognese
Maciej Barylka	4	1	Pizza Hawajska
Krzysztof Morawski	4	4	Pizza Serowa
Krzysztof Morawski	5	3	Pizza Prosciutto

6.2. Przykładowa faktura – zrzut ekranu raportu PDF

Faktura została utworzona za pomocą programu JasperReports, zawiera dwa sub-zapytania odpowiedzialne za dane klienta. Faktura ta dołączona jest do dokumentacji w folderze „raporty” wraz z plikiem programu „JasperReports”. Zawiera ona numer zamówienia, numer faktury, datę wystawienia faktury oraz złożenia zamówienia, dane kupującego oraz poszczególne pozycje zamówienia wraz z kwotami brutto oraz netto zakupu.

- W pierwszym kroku podajemy numer zamówienia dla którego chcemy wygenerować fakturę



Parameter prompt

You can provide a value for the parameter:


id_zamowienia

13

The class type is: java.lang.Integer

OK Use default

- Następnie program generuje fakturę dla zadanego przez nas numeru zamówienia

 <p>FIRMA SPRZEDAJĄCA: Pizzeria Italiano ul. Południowa 130 08-110 Siedlce</p>	FAKTURA		Data wystawienia: niedziela 05 luty 2017 01:02																																						
	nr 13/2017		Data złożenia zamówienia: niedziela 29 styczeń 2017 16:01																																						
DLA NUMERU ZAMÓWIENIA: 13			DANE KUPUJĄCEGO:																																						
			Michał Jaworski Pomorska 3/23 08-110 Siedlce																																						
<table border="1"><thead><tr><th>POZYCJA</th><th>NAZWA DANIA</th><th>LICZBA</th><th colspan="2">CENA ZAKUPU BRUTTO</th><th colspan="2">CENA ZAKUPU NETTO</th></tr></thead><tbody><tr><td>1</td><td>Pizza Margeritha</td><td>1</td><td>14.5</td><td>PLN</td><td>11.31</td><td>PLN</td></tr><tr><td>2</td><td>Pizza Prosciutto</td><td>3</td><td>53.4</td><td>PLN</td><td>41.65</td><td>PLN</td></tr><tr><td>3</td><td>Spaghetti Bolognese</td><td>2</td><td>31.98</td><td>PLN</td><td>24.94</td><td>PLN</td></tr><tr><td colspan="3"></td><td colspan="2">RAZEM BRUTTO:</td><td>99.88</td><td>PLN</td></tr></tbody></table>							POZYCJA	NAZWA DANIA	LICZBA	CENA ZAKUPU BRUTTO		CENA ZAKUPU NETTO		1	Pizza Margeritha	1	14.5	PLN	11.31	PLN	2	Pizza Prosciutto	3	53.4	PLN	41.65	PLN	3	Spaghetti Bolognese	2	31.98	PLN	24.94	PLN				RAZEM BRUTTO:		99.88	PLN
POZYCJA	NAZWA DANIA	LICZBA	CENA ZAKUPU BRUTTO		CENA ZAKUPU NETTO																																				
1	Pizza Margeritha	1	14.5	PLN	11.31	PLN																																			
2	Pizza Prosciutto	3	53.4	PLN	41.65	PLN																																			
3	Spaghetti Bolognese	2	31.98	PLN	24.94	PLN																																			
			RAZEM BRUTTO:		99.88	PLN																																			
<p>Pizzeria Italiano ul. Południowa 130 08-110 Siedlce</p> <p>PODPIS WYSTAWIAJĄCEGO:</p>																																									