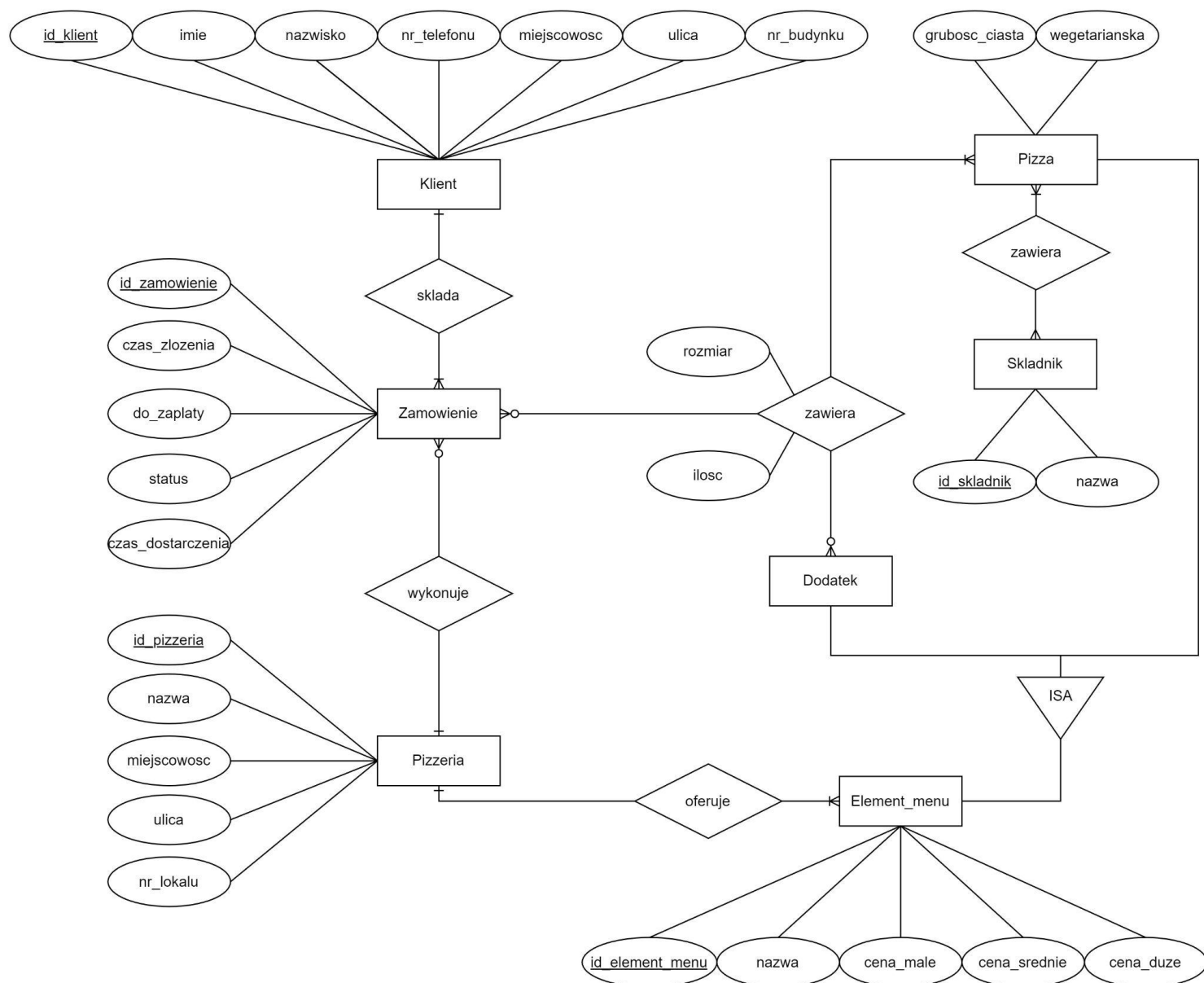


1. Opis problemu

Celem projektu było stworzenie oraz zaimplementowanie w MySQLu bazy danych strony internetowej pizzerii, na której klienci będą mieli możliwość składania zamówień. Tabela *Klient* przechowuje dane użytkowników strony składających zamówienie, takie jak adres oraz dane osobowe. Strona ta miała naśladować schemat takich portali jak np. "pizza.pl", co za tym idzie, powstała baza umożliwia dokonanie wyboru pizzerii z pośród pewnej ich puli. Na tabelę *Pizzeria* składają się informacje adresowe restauracji oraz ich nazwy i ID. Klient posiada możliwość sprecyzowania rozmiaru oraz ilości elementów zamówienia w tabelach *Zamowienie_Pizza* oraz *Zamowienie_Dodatek*. Status zamówienia, czas jego złożenia i dostarczenia oraz sumę do zapłaty znaleźć można w tabeli *Zamowienie*. Projekt bierze również pod uwagę potrzebę przechowywania *Elementow_Menu* każdej pizzerii wraz z ich cenami (zależnymi od rozmiaru), jak i składników składających się na dane pizze (*Pizza_Skladnik*), oraz innych informacji dotyczących oferty pizzerii.

2. Diagram



3. Model relacyjny

1. Klient (id_klient, imie, nazwisko, nr_telefonu, miejscowosc, ulica, nr_budynku)
2. Pizzeria (id_pizzeria, nazwa, miejscowosc, ulica, nr_lokalu)
3. Zamowienie (id_zamowienie, id_klient, id_pizzeria, czas_zlozenia, do_zaplaty, status_zam, czas_dostarczenia)
4. Element_menu (id_element_menu, nazwa, cena_male, cena_srednie, cena_duze, id_pizzeria)
5. Pizza(id_pizza, nazwa, cena_male, cena_srednie, cena_duze, grubosc_ciasta, wegetarianska, id_pizzeria)
6. Dodatek(id_dodatek, nazwa, cena_male, cena_srednie, cena_duze, id_pizzeria)
7. Skladnik (id_skladnik, nazwa)
8. Pizza_Skladnik (id_pizza, id_skladnik)
9. Zamowienie_Pizza (id_zamowienie, id_pizza, rozmiar, ilosc)
10. Zamowienie_Dodatek (id_zamowienie, id_dodatek, rozmiar, ilosc)

4. Implementacja w MySQLu

4.1 Klient

CREATE TABLE Klient

```
(
  id_klient INT NOT NULL AUTO_INCREMENT,
  imie VARCHAR(30) NOT NULL,
  nazwisko VARCHAR(30) NOT NULL,
  nr_telefonu VARCHAR(11) NOT NULL,
  miejscowosc VARCHAR(50) NOT NULL,
  ulica VARCHAR(50) NOT NULL,
  nr_budynku INT NOT NULL,
  PRIMARY KEY(id_klient)
);
```

#klucz (id_klient) nie może być pusty, ponieważ na podstawie niego identyfikujemy wiersz
#wartość id_klient automatycznie wzrasta z każdym nowym rekordem
#wszystkie kolumny są NOT NULL, ponieważ bez nich nie byłyby możliwe dalsze akcje dotyczące zamówienia

```
INSERT INTO `klient` (`id_klient`, `imie`, `nazwisko`, `nr_telefonu`, `miejscowosc`, `ulica`, `nr_budynku`) VALUES
(NULL, 'Julia', 'Duch', '786 123 090', 'Kraków', 'Centralna', '52'),
(NULL, 'Hubert', 'Biliński', '332 145 229', 'Brzesko', 'Różana', '1'),
(NULL, 'Amelia', 'Kmieć', '834 221 090', 'Tarnów', 'Wałowa', '18'),
(NULL, 'Łukasz', 'Kaszowski', '666 983 03', 'Kraków', 'Słomiana', '23'),
(NULL, 'Małgorzata', 'Bielik', '069 420 860', 'Tarnów', 'Marynarki Wojennej', '98'),
(NULL, 'Robert', 'Zieliński', '198 223 601', 'Kraków', 'Zachodnia', '44');
```

4.2 Pizzeria

CREATE TABLE Pizzeria

```
(
  id_pizzeria INT NOT NULL AUTO_INCREMENT,
```

```

nazwa VARCHAR(50) NOT NULL,
mijscowosc VARCHAR(40) NOT NULL,
ulica VARCHAR(40) NOT NULL,
nr_lokalu INT NOT NULL,
PRIMARY KEY(id_pizzeria)
);
#klucz (id_pizzeria) nie może być pusty, ponieważ na podstawie niego identyfikujemy wiersz
#wartość klucza automatycznie wzrasta z każdym nowym rekordem
#wszystkie informacje są niezbędne, dlatego każdy atrybut ma wartość NOT NULL

```

```

INSERT INTO `pizzeria` (`id_pizzeria`, `nazwa`, `mijscowosc`, `ulica`, `nr_lokalu`) VALUES
(NULL, 'PizzaHut', 'Kraków', 'Drukarska', '18'),
(NULL, 'Toni Pepperoni', 'Kraków', 'Miłkowskiego', '17'),
(NULL, 'Pizzeria Soprano', 'Brzesko', 'Kopernika', '9'),
(NULL, 'Pizzeria Etna', 'Tarnów', 'Kraśńskiego', '12'),
(NULL, 'Pizzeria Verona', 'Tarnów', 'Akacyjowa', '1'),
(NULL, 'Maxi Pizza', 'Kraków', 'Kapelanka', '15');

```

4.3 Element_menu

```

CREATE TABLE Element_menu
(
    id_element_menu INT NOT NULL AUTO_INCREMENT,
    nazwa VARCHAR(40) NOT NULL,
    cena_male DECIMAL(5,2),
    cena_srednie DECIMAL(5,2),
    cena_duze DECIMAL(5,2),
    id_pizzeria INT NOT NULL,
    PRIMARY KEY(id_element_menu),
    FOREIGN KEY(id_pizzeria) REFERENCES Pizzeria(id_pizzeria)
);
#klucz (id_element_menu) nie może być pusty, ponieważ na podstawie niego identyfikujemy wiersz
#wartość klucza automatycznie wzrasta z każdym nowym rekordem
#ceny nie są NOT NULL, ponieważ nie wszystkie produkty muszą mieć każdy rozmiar
#do zapisu cen użyty został typ danych DECIMAL, ponieważ FLOAT może przechowywać
wartości o różnej precyzji. Wartości w FLOAT może być zarówno 1.6 jak i 1.6180339887, co
mogłoby stanowić problem, np. w przypadku faktur.

```

```

INSERT INTO `element_menu` (`id_element_menu`, `nazwa`, `cena_male`, `cena_srednie`,
`cena_duze`, `id_pizzeria`) VALUES
(NULL, 'Margharita', '20.00', '22.00', '24.00', '6'),
(NULL, 'Coca Cola', '5.00', '7.00', '10.00', '1'),
(NULL, 'Grecka', '23.00', '27.00', '29.00', '4'),
(NULL, 'Lemoniada', '3.00', '4.00', '5.00', '4'),
(NULL, 'Vege', '25.00', '27.50', '29.00', '5'),
(NULL, 'Woda', '3.00', '4.00', '5.00', '3'),
(NULL, 'Pepperoni', '24.00', '25.00', '26.50', '2'),
(NULL, 'Sos czosnkowy', '2.00', '3.00', '4.00', '1'),
(NULL, 'Sos czosnkowy', '3.00', '4.00', '5.50', '3'),

```

```
(NULL, 'Brokułowa', '23.00', '25.00', '28.00', '5'),  
(NULL, 'Mięsna', '28.00', '30.00', '33.00', '6'),  
(NULL, 'Sok jabłkowy', '4.50', '6.00', '7.50', '2');
```

4.4 Składnik

```
CREATE TABLE Skladnik
```

```
(  
    id_skladnik INT NOT NULL AUTO_INCREMENT,  
    nazwa VARCHAR(40) NOT NULL,  
    PRIMARY KEY(id_skladnik)  
);
```

#klucz nie może być pusty, ponieważ na podstawie niego identyfikujemy wiersz

#wartość klucza automatycznie wzrasta z każdym nowym rekordem

```
INSERT INTO `skladnik` (`id_skladnik`, `nazwa`) VALUES
```

```
(NULL, 'Szynka'),  
(NULL, 'Kurczak'),  
(NULL, 'Brokuł'),  
(NULL, 'Ser'),  
(NULL, 'Sos pomidorowy'),  
(NULL, 'Mozzarella'),  
(NULL, 'Papryka'),  
(NULL, 'Cebula'),  
(NULL, 'Kukurydza'),  
(NULL, 'Pomidor'),  
(NULL, 'Szpinak'),  
(NULL, 'Feta'),  
(NULL, 'Oliwki'),  
(NULL, 'Pepperoni'),  
(NULL, 'Boczek'),  
(NULL, 'Szynka parmeńska'),  
(NULL, 'Bazylia'),  
(NULL, 'Pieczarki'),  
(NULL, 'Papryczka ostra');
```

4.5 Pizza

```
CREATE TABLE Pizza
```

```
(  
    id_pizza INT NOT NULL AUTO_INCREMENT,  
    nazwa VARCHAR(40) NOT NULL,  
    cena_male DECIMAL(5,2),  
    cena_srednie DECIMAL(5,2),  
    cena_duze DECIMAL(5,2),  
    grubosc_ciesta VARCHAR(30) NOT NULL,  
    wegetarianska CHAR(3) NOT NULL,  
    id_pizzeria INT NOT NULL,  
    PRIMARY KEY(id_pizza),
```

```

FOREIGN KEY(id_pizza) REFERENCES Element_menu(id_element_menu),
FOREIGN KEY(id_pizzeria) REFERENCES Pizzeria(id_pizzeria)
);
#klucz nie może być pusty, ponieważ na podstawie niego identyfikujemy wiersz
#wartość klucza automatycznie wzrasta z każdym nowym rekordem
#atrybut wegetarianska zapisywana jest w typie CHAR(3), ponieważ wpisaną tam daną
może być jedynie tak lub nie

```

```

INSERT INTO `pizza` (`id_pizza`, `nazwa`, `cena_male`, `cena_srednie`, `cena_duze`,
`grubosc_ciesta`, `wegetarianska`, `id_pizzeria`) VALUES
('1', 'Margharita', '20.00', '22.00', '24.00', 'grube', 'tak', '6'),
('3', 'Grecka', '23.00', '27.00', '29.00', 'cienkie', 'tak', '4'),
('5', 'Vege', '25.00', '27.50', '29.00', 'grube', 'tak', '5'),
('7', 'Pepperoni', '24.00', '25.00', '26.50', 'grube', 'nie', '2'),
('10', 'Brokułowa', '23.00', '25.00', '28.00', 'cienkie', 'nie', '5'),
('11', 'Mięsna', '28.00', '30.00', '33.00', 'cienkie', 'nie', '6');

```

4.6 Dodatek

```

CREATE TABLE Dodatek
(
    id_dodatek INT NOT NULL AUTO_INCREMENT,
    nazwa VARCHAR(40) NOT NULL,
    cena_male DECIMAL(5,2),
    cena_srednie DECIMAL(5,2),
    cena_duze DECIMAL(5,2),
    id_pizzeria INT NOT NULL,
    PRIMARY KEY(id_dodatek),
    FOREIGN KEY(id_dodatek) REFERENCES Element_menu(id_element_menu),
    FOREIGN KEY(id_pizzeria) REFERENCES Pizzeria(id_pizzeria)
);
#klucz nie może być pusty, ponieważ na podstawie niego identyfikujemy wiersz
#wartość klucza automatycznie wzrasta z każdym nowym rekordem

```

```

INSERT INTO `dodatek` (`id_dodatek`, `nazwa`, `cena_male`, `cena_srednie`, `cena_duze`,
`id_pizzeria`) VALUES
('2', 'Coca Cola', '5.00', '7.00', '10.00', '1'),
('4', 'Lemoniada', '3.00', '4.00', '5.00', '4'),
('6', 'Woda', '3.00', '4.00', '5.00', '3'),
('8', 'Sos czoskowy', '2.00', '3.00', '4.00', '1'),
('9', 'Sos czosnkowy', '3.00', '4.00', '5.50', '3'),
('12', 'Sok jabłkowy', '4.50', '6.00', '7.50', '2');

```

4.7 Pizza i Sładnik

```

CREATE TABLE Pizza_Skladnik
(
    id_pizza INT NOT NULL,
    id_skladnik INT NOT NULL,

```

```

PRIMARY KEY(id_pizza, id_skladnik),
FOREIGN KEY(id_pizza) REFERENCES Pizza(id_pizza),
FOREIGN KEY(id_skladnik) REFERENCES Skladnik(id_skladnik)
);
#klucz nie może być pusty, ponieważ na podstawie niego identyfikujemy wiersz
#wartość klucza automatycznie wzrasta z każdym nowym rekordem

```

```

INSERT INTO `pizza_skladnik` (`id_pizza`, `id_skladnik`) VALUES
('1', '4'), ('1', '5'), ('1', '6'), ('1', '17'), ('3', '5'), ('3', '12'), ('3', '13'), ('3', '10'), ('5', '5'), ('5', '4'), ('5', '18'),
('5', '11'), ('7', '5'), ('7', '14'), ('7', '19'), ('7', '8'), ('10', '3'), ('10', '2'), ('10', '4'), ('10', '1'), ('11', '4'), ('11',
'15'), ('11', '16'), ('11', '19');

```

4.8 Zamówienie

```

CREATE TABLE Zamowienie
(
    id_zamowienie INT NOT NULL AUTO_INCREMENT,
    id_klient INT NOT NULL,
    id_pizzeria INT NOT NULL,
    czas_zlozenia DATETIME NOT NULL,
    do_zaplaty INT NOT NULL,
    status_zam VARCHAR(40) NOT NULL,
    czas_dostarczenia DATETIME,
    PRIMARY KEY(id_zamowienie),
    FOREIGN KEY(id_klient) REFERENCES Klient(id_klient),
    FOREIGN KEY(id_pizzeria) REFERENCES Pizzeria(id_pizzeria)
);
#klucz nie może być pusty, ponieważ na podstawie niego identyfikujemy wiersz
#wartość klucza automatycznie wzrasta z każdym nowym rekordem
#atrybut czas_dostarczenia jest przechowywany w typie DATETIME, ponieważ potrzebna jest nam
zarówno data, jak i godzina

```

```

INSERT INTO `zamowienie` (`id_zamowienie`, `id_klient`, `id_pizzeria`, `czas_zlozenia`,
`do_zaplaty`, `status_zam`, `czas_dostarczenia`) VALUES
(NULL, '6', '1', '2022-04-05 20:49:21', '30.00', 'dostarczone', '2022-04-05 21:20:25'),
(NULL, '5', '2', '2022-05-28 22:40:25', '57.50', 'w drodze', NULL),
(NULL, '4', '3', '2022-05-11 18:59:54', '9.00', 'dostarczone', '2022-05-11 19:15:04'),
(NULL, '3', '4', '2022-04-21 18:32:29', '33.00', 'dostarczone', '2022-04-21 19:02:48'),
(NULL, '2', '5', '2022-04-25 18:55:02', '105.50', 'dostarczone', '2022-04-25 19:43:12'),
(NULL, '1', '6', '2022-05-10 21:07:33', '52', 'dostarczone', '2022-05-10 22:01:53');

```

4.9 Zamówiona Pizza

```

CREATE TABLE Zamowienie_Pizza
(
    id_zamowienie INT NOT NULL,
    id_pizza INT NOT NULL,
    ilosc INT NOT NULL,
    rozmiar VARCHAR(30) NOT NULL,

```

```

PRIMARY KEY(id_zamowienie, id_pizza),
FOREIGN KEY(id_pizza) REFERENCES Pizza(id_pizza),
FOREIGN KEY(id_zamowienie) REFERENCES Zamowienie(id_zamowienie)
);
#klucz nie może być pusty, ponieważ na podstawie niego identyfikujemy wiersz
#wartość klucza automatycznie wzrasta z każdym nowym rekordem
#atrybuty związku "zawiera" trafiają do tej tabelki

INSERT INTO `zamowienie_pizza` (`id_zamowienie`, `id_pizza`, `ilosc`, `rozmiar`) VALUES ('2',
'7', '2', 'duża'),
('4', '3', '1', 'mała'),
('5', '5', '3', 'średnia'),
('5', '10', '1', 'mała'),
('6', '1', '1', 'średnia'),
('6', '11', '1', 'średnia');

```

4.10 Zamówiony Dodatek

```

CREATE TABLE Zamowienie_Dodatek
(
    id_zamowienie INT NOT NULL,
    id_dodatek INT NOT NULL,
    ilosc INT NOT NULL,
    rozmiar VARCHAR(30) NOT NULL,
    PRIMARY KEY(id_zamowienie, id_dodatek),
    FOREIGN KEY(id_dodatek) REFERENCES Dodatek(id_dodatek),
    FOREIGN KEY(id_zamowienie) REFERENCES Zamowienie(id_zamowienie)
);
#klucz nie może być pusty, ponieważ na podstawie niego identyfikujemy wiersz
#wartość klucza automatycznie wzrasta z każdym nowym rekordem
#atrybuty relacji zawiera trafiają do tej tabelki

INSERT INTO `zamowienie_dodatek` (`id_zamowienie`, `id_dodatek`, `ilosc`, `rozmiar`) VALUES
('1', '2', '3', 'duży'),
('2', '12', '1', 'mały'),
('3', '6', '1', 'duży'),
('3', '9', '1', 'średni'),
('4', '4', '2', 'duży');

```

5. Postać normalna

5.1 Klient

Atrybut `id_klient` jest kluczem głównym relacji, posiada wartości unikalne. Zbiór atrybutów {*imie*, *nazwisko*, *nr_telefonu*, *miestowosc*, *ulica*, *nr_budynku*} mogą się powielać.

Jedyna nietrywialna zależność funkcyjna relacji *Klient* to:

`id_klient` → `imie`, `nazwisko`, `nr_telefonu`, `miestowosc`, `ulica`, `nr_budynku`

Relacja spełnia wymagania 1NF - opisuje jeden obiekt, wartości atrybutów są atomowe, nie mogą w niej występować dwa jednakowe wiersze.

Relacja spełnia wymagania 2NF - jest w 1NF, wszystkie atrybuty niekluczowe są w zależności funkcyjnej od klucza.

Relacja spełnia wymagania 3NF - jest w 2NF, żaden atrybut niekluczowy nie jest zależny funkcyjnie od innych atrybutów niekluczowych.

Relacja spełnia wymagania BCNF - jest w 3NF, każdy atrybut w tabeli, od którego w pełni funkcyjnie zależy inny atrybut, musi być kluczem kandydującym.

5.2 Pizzeria

Jedyna nietrywialna zależność funkcyjna relacji *Pizzeria* to:

$\text{id_pizzeria} \rightarrow \text{nazwa, miejscowosc, ulica, nr_budynku}$

Relacja spełnia warunki 1NF, 2NF, 3NF, BCNF.

5.3 Element_menu

Jedynie nietrywialne zależności funkcyjne relacji *Element_menu* to:

$\text{id_element_menu} \rightarrow \text{id_pizzeria, nazwa, cena_male, cena_srednie, cena_duze}$

$\text{id_pizzeria, nazwa} \rightarrow \text{id_element_menu, cena_male, cena_srednie, cena_duze}$

$\{\text{id_element_menu}\}^+$ oraz $\{\text{id_pizzeria, nazwa}\}^+$ zawierają wszystkie atrybuty tej relacji (są nadkluczami), tabela spełnia więc warunki 1NF, 2NF, 3NF, BCNF.

5.4 Skladnik

Jedyna nietrywialna zależność funkcyjna relacji *Skladnik* to:

$\text{id_skladnik} \rightarrow \text{nazwa}$

Tabela spełnia więc warunki 1NF, 2NF, 3NF, BCNF.

5.5 Pizza

Jedynie nietrywialne zależności funkcyjne relacji *Pizza* to:

$\text{id_pizza} \rightarrow \text{nazwa, cena_male, cena_srednie, cena_duze, grubosc_ciasta, wegetarianska, id_pizzeria}$

$\text{id_pizzeria, nazwa} \rightarrow \text{id_pizza, cena_male, cena_srednie, cena_duze, grubosc_ciasta, wegetarianska}$

$\{\text{id_pizza}\}^+$ oraz $\{\text{id_pizzeria, nazwa}\}^+$ zawierają wszystkie atrybuty tej relacji (są nadkluczami), tabela spełnia więc warunki 1NF, 2NF, 3NF, BCNF.

5.6 Dodatek

Jedynie nietrywialne zależności funkcyjne relacji *Dodatek* to:

$\text{id_dodatek} \rightarrow \text{nazwa, cena_male, cena_srednie, cena_duze, id_pizzeria}$

$\text{id_pizzeria, nazwa} \rightarrow \text{id_dodatek, cena_male, cena_srednie, cena_duze}$

$\{\text{id_dodatek}\}^+$ oraz $\{\text{id_pizzeria, nazwa}\}^+$ zawierają wszystkie atrybuty tej relacji (są nadkluczami), tabela spełnia więc warunki 1NF, 2NF, 3NF, BCNF.

5.7 Pizza_Skladnik

Brak relacji nietrywialnych, tabela spełnia więc warunki 1NF, 2NF, 3NF, BCNF.

5.8 Zamowienie

Jedynie nietrywialne zależności funkcyjne relacji *Zamowienie* to:

$\text{id_zamowienie} \rightarrow \text{id_klient, id_pizzeria, czas_zlozenia, do_zaplaty, status_zam, czas_dostarczenia}$

$\text{id_klient, id_pizzeria, czas_zlozenia} \rightarrow \text{id_zamowienie, do_zaplaty, status_zam, czas_dostarczenia}$

$\{\text{id_zamowienie}\}^+$ oraz $\{\text{id_klient, id_pizzeria, czas_zlozenia}\}^+$ zawierają wszystkie atrybuty tej relacji (są nadkluczami), tabela spełnia więc warunki 1NF, 2NF, 3NF, BCNF.

5.9 Zamowienie_Pizza

Jedyna nietrywialna zależność funkcyjna relacji *Zamowienie_Pizza* to:

$\text{id_zamowienie id_pizza} \rightarrow \text{rozmiar, ilosc}$

Tabela spełnia więc warunki 1NF, 2NF, 3NF, BCNF.

5.10 Zamowienie_Dodatek

Jedyna nietrywialna zależność funkcyjna relacji *Zamowienie_Dodatek* to:

$\text{id_zamowienie, id_dodatek} \rightarrow \text{rozmiar, ilosc}$

Tabela spełnia więc warunki 1NF, 2NF, 3NF, BCNF.