

## Sprawozdanie 4

# Projektowanie danych.

### Implementacyjny diagram klas

Diagram klas implementacyjnych przedstawia strukturę logiczną systemu w języku programowania. Główne klasy to:

- **Zamowienie** – łączy informacje o kliencie, produktach, płatności, statusie i fakturze.
- **Klient** – klasa ogólna z podklasami KlientIndywidualny oraz KlientFirmowy.
- **Produkt** – element oferty, zawarty w pozycjach zamówienia.
- **PozycjaZamowienia** – pośrednia klasa łącząca Zamowienie i Produkt.
- **PracownikObslugi**, Kucharz, Administrator – klasy reprezentujące użytkowników z różnymi uprawnieniami.
- **Magazyn, Dostawa, Raport, Platnosc, Faktura** – pozostałe elementy systemu wspierające zarządzanie.

Dziedziczenie oraz powiązania wiele-do-wielu zostały odpowiednio odwzorowane.

### Opis tabel

- **zamowienie** – zawiera informacje o czasie realizacji, powiązaniach z klientem, pracownikiem i kucharzem.
- **klient** – wspólna tabela dla wszystkich klientów; szczegółowe dane dla firmowych w osobnej tabeli.
- **produkt** – zawiera dane o nazwie, cenie i kategorii produktu.

- **pozycja\_zamowienia** – realizuje relację wiele-do-wielu między zamówieniem a produktami.
- **platnosc** – określa sposób i kwotę zapłaty, powiązana z konkretnym zamówieniem.

## Diagram relacyjnej bazy danych

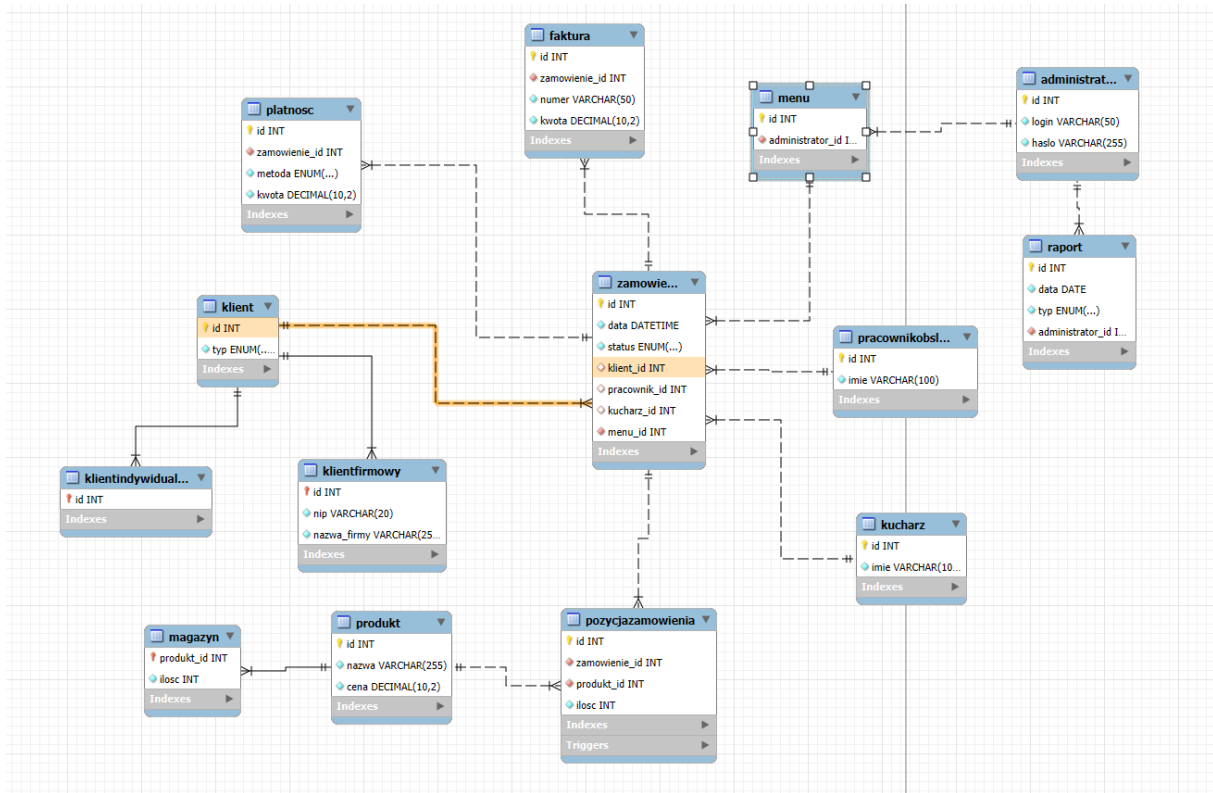


Diagram relacyjnej bazy danych przedstawia zależności między tabelami systemu. Najważniejsze tabele to:

- **klient** (id, imie, nazwisko, typ, ...)
- **klient\_firmowy** (id\_klienta, nazwa\_firmy, nip)
- **zamowienie** (id, data, status, id\_klienta, id\_pracownika, id\_kucharza)
- **produkt** (id, nazwa, cena, kategoria)
- **pozycja\_zamowienia** (id\_zamowienia, id\_produktu, ilosc)
- **platnosc** (id, metoda, kwota, id\_zamowienia)
- **faktura** (id, id\_zamowienia, data, kwota\_brutto)

- **pracownik** (id, imie, rola)
- **magazyn** (id\_produktu, ilosc)
- **dostawa** (id, data, id\_produktu, ilosc)
- **raport** (id, typ, zakres\_czasowy, data\_generowania)

## **Normalizacja**

Baza została zaprojektowana zgodnie z zasadami trzeciej postaci normalnej (3NF):

- wszystkie dane są atomowe,
- nie występują redundancje,
- zależności między kolumnami są logicznie uzasadnione,
- dane powtarzalne (np. statusy, role, metody płatności) mogą być przeniesione do tabel słownikowych

## **Mapowanie klas na tabele**

Diagram klas implementacyjnych został zmapowany na relacyjną bazę danych w następujący sposób:

- Każda klasa odpowiada jednej tabeli (np. Zamowienie → zamowienie).
- Dziedziczenie Klient → KlientFirmowy i KlientIndywidualny zostało zaimplementowane przez wspólny klucz główny id.
- Relacje między klasami, np. Zamowienie – PozycjaZamowienia – Produkt, są odwzorowane przez tabele pośrednie z kluczami obcymi.

## **Typy danych i ograniczenia**

Przykładowe typy danych użyte w projekcie:

- INT, VARCHAR, DATE, DECIMAL(6,2), ENUM('gotówka','karta','blik')
- Klucze główne: PRIMARY KEY

- Klucze obce: FOREIGN KEY
- Ograniczenia integralności: NOT NULL, CHECK, UNIQUE

## **Podsumowanie**

Zaprojektowana relacyjna baza danych stanowi logiczną kontynuację wcześniejszych etapów projektowania systemu. Zawarte w niej tabele odwzorowują rzeczywiste obiekty systemowe, a zaplanowane relacje i ograniczenia zapewniają spójność danych. Baza jest gotowa do implementacji w środowisku produkcyjnym i może stanowić podstawę dla systemu obsługi zamówień, magazynu i raportowania.

### **Autorzy:**

Jakub Góralski

Anastazja Albeszczenko