

# Projekt Bazy Danych

Antoni Konecko 130528  
Bioinformatyka

## Wprowadzenie - Wybranie dziedziny rzeczywistości i opis

### **Projekt bazy danych sklepu internetowego.**

Baza zawiera informacje o klientach, zamówieniach, fakturach, produktach oraz pracownikach.

Klient rejestrując się dostaje unikalne id, które wraz z podanym przez klienta hasłem (w bazie zapisywany jest 60 znakowy hash z hasła użytkownika wygenerowany przez algorytm hashujący np. Argon2) służą do logowania, Klient musi także podać imię, nazwisko, adres, nr telefonu oraz email. W przypadku gdy klientem jest firma można podać NIP oraz nazwę firmy.

Każdy klient może stworzyć wiele zamówień, a dane zamówienie może być utworzone tylko przez jednego klienta. Zamówienie musi zawierać informacje o unikalnym id zamówienia, dacie jego złożenia, o tym czy zostało przyjęte i zrealizowane oraz opcjonalnie o dacie przyjęcia, o tym czy jest opłacone, a także dacie wysyłki i dacie realizacji.

Każde zamówienie musi zawierać jeden lub więcej produktów a produkty mogą znaleźć się w wielu zamówieniach.

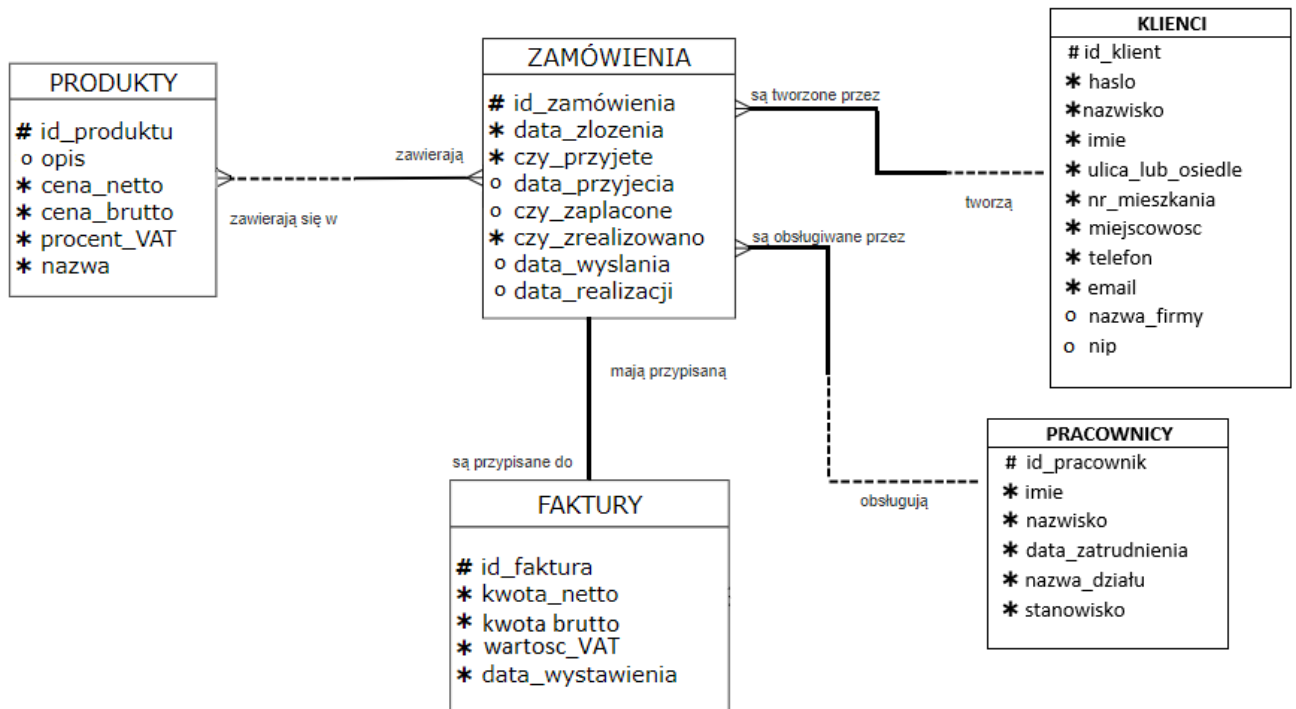
Produkty przechowują informację o id produktu, nazwie produktu, cenie netto, brutto, procencie VAT i opcjonalny opis produktu.

Każde zamówienie ma przypisaną fakturę, która zawiera informacje id faktury, kwocie netto, kwocie brutto, wartości VAT oraz dacie wystawienia.

Każde zamówienie jest obsługiwane przez konkretnego pracownika a jeden pracownik może obsługiwać wiele zamówień.

Przechowywane informacje o pracownikach to : nadane id, imię, nazwisko, data zatrudnienia, nazwa działu i stanowisko.

## Model związków encji EER ( notacja Barkera )



## Schemat relacyjnej bazy danych

KLIENCI:

```

CREATE TABLE klienci (
    id_klient          INTEGER
        GENERATED ALWAYS AS IDENTITY
        PRIMARY KEY,

    haslo              CHAR(60)    NOT NULL,
    nazwisko           VARCHAR(20) NOT NULL,
    imie               VARCHAR(20) NOT NULL,
    ulica_lub_osiedle  VARCHAR(60) NOT NULL,
    nr_mieszkania       VARCHAR(10) NOT NULL,
    miejscowosc        VARCHAR(30) NOT NULL,
    telefon            VARCHAR(9)  NOT NULL,
    email              VARCHAR(30) NOT NULL,
    nazwa_firmy        VARCHAR(30),
    nip                VARCHAR(10)
);
  
```

## PRACOWNICY:

```
CREATE TABLE pracownicy (  
    id_pracownik      INTEGER  
        GENERATED ALWAYS AS IDENTITY  
    PRIMARY KEY,  
  
    imie               VARCHAR(20) NOT NULL,  
    nazwisko           VARCHAR(20) NOT NULL,  
    data_zatrudnienia DATE        NOT NULL,  
    nazwa_dzialu       VARCHAR(20) NOT NULL,  
    stanowisko         VARCHAR(20) NOT NULL  
);
```

## PRODUKTY:

```
CREATE TABLE produkty (  
    id_produktu      INTEGER  
        GENERATED ALWAYS AS IDENTITY  
    PRIMARY KEY,  
  
    opis             VARCHAR(200),  
    cena_netto       DECIMAL(5, 2) NOT NULL,  
    cena_brutto      DECIMAL(5, 2) NOT NULL,  
    procent_vat       DECIMAL(4, 3) NOT NULL,  
    nazwa            VARCHAR(20) NOT NULL,  
    CONSTRAINT chk_ceny CHECK (cena_netto > 0 AND procent_vat > 0 AND  
                                cena_netto < cena_brutto AND  
                                cena_netto + (cena_netto * procent_vat) = cena_brutto)  
);
```

## FAKTURY:

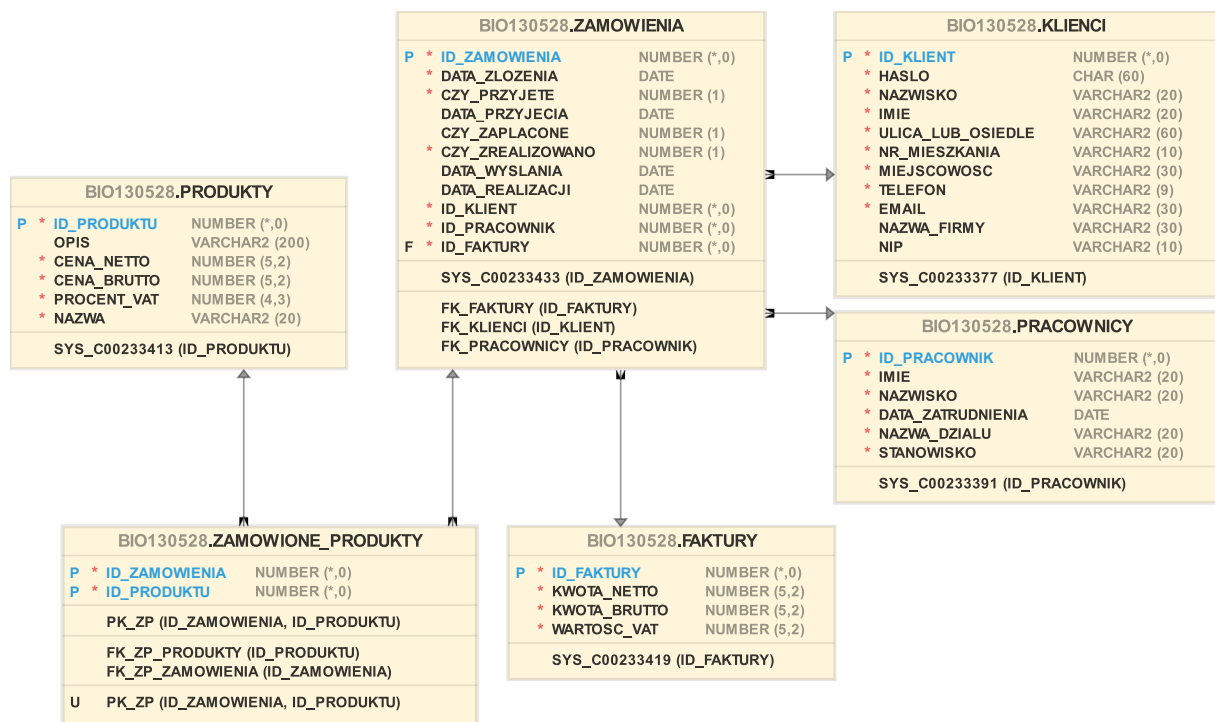
```
CREATE TABLE faktury ( id_faktury INTEGER  
    GENERATED ALWAYS AS IDENTITY  
    PRIMARY KEY,  
  
    kwota_netto       DECIMAL(5, 2) NOT NULL,  
    kwota_brutto      DECIMAL(5, 2) NOT NULL,  
    wartosc_vat       DECIMAL(5, 2) NOT NULL,  
  
    CONSTRAINT chk_faktury CHECK ( kwota_netto > 0 AND wartosc_vat > 0 AND  
                                    kwota_netto < kwota_brutto AND  
                                    kwota_netto + wartosc_vat = kwota_brutto)  
);
```

## ZAMÓWIENIA:

```
CREATE TABLE zamowienia (  
    id_zamowienia      INTEGER  
        GENERATED ALWAYS AS IDENTITY  
    PRIMARY KEY,  
  
    data_zlozenia      DATE DEFAULT CURRENT_DATE NOT NULL,  
    czy_przyjete       boolean NOT NULL,  
    data_przyjecia     DATE,  
    czy_zaplacone      boolean,  
    czy_zrealizowano  boolean NOT NULL,  
    data_wyslania      DATE,  
    data_realizacji    DATE,  
  
    id_klient          INTEGER  
        CONSTRAINT fk_klienci  
        REFERENCES klienci ( id_klient )  
    NOT NULL,  
    id_pracownik       INTEGER  
        CONSTRAINT fk_pracownicy  
        REFERENCES pracownicy ( id_pracownik )  
    NOT NULL,  
    id_faktury         INTEGER  
        CONSTRAINT fk_faktury  
        REFERENCES faktury ( id_faktury )  
    NOT NULL,  
  
    CONSTRAINT chk_daty CHECK (data_zlozenia <= data_przyjecia AND  
                                data_przyjecia <= data_wyslania AND  
                                data_wyslania <= data_realizacji)  
  
);
```

## ZAMOWIONE\_PRODUKTY:

```
CREATE TABLE zamowione_produkty (  
    id_zamowienia      INTEGER  
        CONSTRAINT fk_zp_zamowienia  
        REFERENCES zamowienia(id_zamowienia) NOT NULL,  
    id_produktu        INTEGER  
        CONSTRAINT fk_zp_produkty  
        REFERENCES produkty(id_produktu) NOT NULL,  
  
    CONSTRAINT pk_zp PRIMARY KEY (id_zamowienia, id_produktu)  
);
```



Relacja ZAMOWIONE\_PRODUKTY jest wynikiem transformacji binarnego związku jednostronnie obowiązkowego M:N encji: ZAMOWIENIA i PRODUKTY.  
Klucz główny tej relacji stanowią oba klucze obce, (wskazujące na klucze główne relacji ZAMÓWIENIA oraz PRODUKTY).

## Normalizacja schematów relacji

Tabele znajdują się w 2 i 3 postaci normalnej:

Żaden atrybut wtórny nie jest częściowo funkcyjnie zależny od żadnego z kluczy relacji oraz żaden atrybut wtórny nie jest przechodnio zależny od klucza głównego relacji. W każdej z relacji klucz główny wyznacza wszystkie atrybuty relacji.