

# Projekt semestralny

## "Aplikacja Bazodanowa"

Piotr Kurczab  
Patryk Kucharz  
Bartosz Wójcik  
Paweł Garbal

1. Opis funkcjonalności aplikacji
2. Struktura bazy danych
  - 2.1. Diagram ERD
  - 2.2. Kod SQL tworzący bazę danych
  - 2.3. Kod SQL wypełniający bazę danych przykładowymi danymi
3. Kod źródłowy aplikacji
4. Testy jednostkowe
5. Dokumentacja kodu
6. Instrukcja instalacji i uruchomienia aplikacji
7. Instrukcja użytkowania aplikacji

## 1. Opis funkcjonalności aplikacji

Aplikacja do zarządzania bazą danych umożliwia użytkownikom przechowywanie i zarządzanie informacjami o klientach, zamówieniach, produktach oraz dostawcach. Główne funkcjonalności aplikacji obejmują:

- Wyświetlanie listy klientów, zamówień, produktów i dostawców w formie tabel
- Filtrowanie i wyszukiwanie danych w tabelach
- Dodawanie nowych rekordów do tabel
- Edycja istniejących rekordów w tabelach
- Usuwanie rekordów z tabel
- Eksportowanie i importowanie danych bazy danych do/z pliku CSV
- Wyświetlanie połączonych danych o zamówieniach klientów wraz z informacjami o klientach i produktach

Interfejs użytkownika jest w języku angielskim. Aplikacja została napisana w języku Python z wykorzystaniem biblioteki PyQt6 dla interfejsu graficznego użytkownika. Do przechowywania danych wykorzystano lekką, wbudowaną w Pythona bazę danych SQLite.

## 2. Struktura bazy danych

Struktura bazy danych składa się z czterech głównych tabel:

### Tabela "customers"

**id:** Unikalny identyfikator klienta (klucz główny, autoinkrementowany)  
**name:** Imię i nazwisko klienta  
**email:** Adres e-mail klienta  
**phone:** Numer telefonu klienta  
**city:** Miasto zamieszkania klienta

### Tabela "orders"

**id:** Unikalny identyfikator zamówienia (klucz główny, autoinkrementowany)  
**customer\_id:** Identyfikator klienta (klucz obcy odnoszący się do tabeli "customers")  
**product\_id:** Identyfikator produktu (klucz obcy odnoszący się do tabeli "products")  
**date:** Data złożenia zamówienia

**amount:** Ilość produktów w zamówieniu  
**status:** Status zamówienia (np. "Wysłane", "W realizacji", "Dostarczone", "Anulowane")

#### Tabela "products"

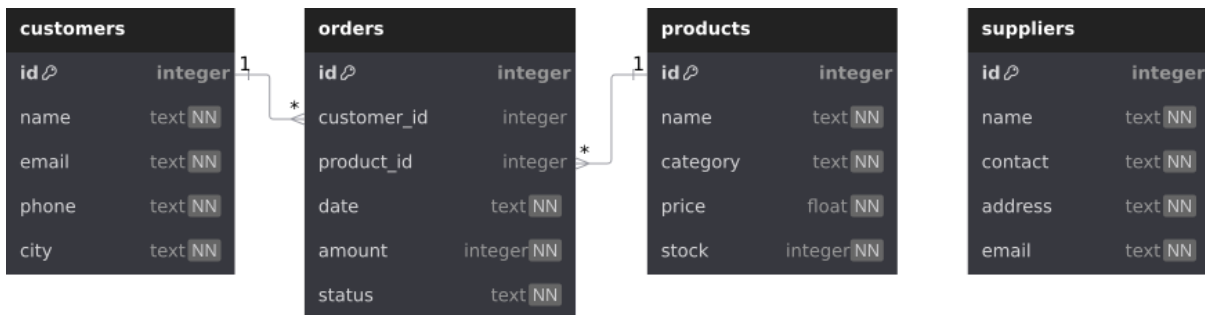
**id:** Unikalny identyfikator produktu (klucz główny, autoinkrementowany)  
**name:** Nazwa produktu  
**category:** Kategoria produktu  
**price:** Cena produktu  
**stock:** Ilość produktu na stanie

#### Tabela "suppliers"

**id:** Unikalny identyfikator dostawcy (klucz główny, autoinkrementowany)  
**name:** Nazwa dostawcy  
**contact:** Osoba kontaktowa u dostawcy  
**address:** Adres dostawcy  
**email:** Adres e-mail dostawcy

Relacje między tabelami są ustalone za pomocą kluczy obcych. Tabela **orders** ma dwa klucze obce: **customer\_id** odnoszący się do tabeli **customers** oraz **product\_id** odnoszący się do tabeli **products**.

### 2.1. Diagram ERD



### 2.2. Kod SQL tworzący bazę danych

```
CREATE TABLE customers (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name TEXT NOT NULL,  
    email TEXT NOT NULL,  
    phone TEXT NOT NULL,  
    city TEXT NOT NULL  
);
```

```
CREATE TABLE orders (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    customer_id INTEGER,  
    product_id INTEGER,  
    date TEXT NOT NULL,  
    amount INTEGER NOT NULL,  
    status TEXT NOT NULL,  
    FOREIGN KEY (customer_id) REFERENCES customers (id),
```

```
FOREIGN KEY (product_id) REFERENCES products (id)
);
```

```
CREATE TABLE products (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    category TEXT NOT NULL,
    price REAL NOT NULL,
    stock INTEGER NOT NULL
);
```

```
CREATE TABLE suppliers (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    contact TEXT NOT NULL,
    address TEXT NOT NULL,
    email TEXT NOT NULL
);
```

## 2.3. Kod SQL wypełniający bazę danych przykładowymi danymi

```
INSERT INTO customers (name, email, phone, city) VALUES
    ("Michał Kowalski", "michal.kowalski@gmail.com", "501-234-567",
    "Warszawa"),
    ("Anna Nowak", "a.nowak@example.com", "694-567-890", "Kraków"),
    ("Piotr Wiśniewski", "p.wisniewski@firma.pl", "789-012-345",
    "Gdańsk"),
    ("Katarzyna Wójcik", "kwojcik@mail.com", "234-567-890",
    "Wrocław");
```

```
INSERT INTO products (name, category, price, stock) VALUES
    ("Smartfon XYZ Pro", "Elektronika", 2499.99, 75),
    ("Laptop GamePro 5000", "Komputery", 4999.00, 20),
    ("Odkurzacz Turbo 2000", "AGD", 799.99, 50),
    ("Zestaw garnków Premium", "Dom i ogród", 599.00, 30);
```

```
INSERT INTO orders (customer_id, product_id, date, amount, status)
VALUES
    (1, 3, "2023-04-22", 1, "Wysłane"),
    (2, 1, "2023-05-11", 2, "W realizacji"),
    (4, 2, "2023-06-03", 1, "Dostarczone"),
    (3, 4, "2023-05-28", 3, "Anulowane");
```

```
INSERT INTO suppliers (name, contact, address, email) VALUES
```

```
("MegaElektro S.A.", "Jan Kowalski", "ul. Przemysłowa 5, 00-123  
Warszawa", "kontakt@megaelektro.pl"),  
("AGDMaster", "Anna Wiśniewska", "ul. Handłowa 27, 80-200  
Gdańsk", "obsługarc@agdmaster.com"),  
("DomBytHouse", "Piotr Zieliński", "ul. Ogrodowa 12, 30-500  
Kraków", "kontakt@dombyt.pl"),  
("TopKomputery", "Katarzyna Adamczyk", "Al. Narodowa 31, 40-100  
Wrocław", "biuro@topkomputery.pl");
```

### 3. Kod źródłowy aplikacji

Kod źródłowy dostępny jest w repozytorium GitHub pod adresem  
[https://github.com/PiotrKurczab/aplikacja\\_bazodanowa](https://github.com/PiotrKurczab/aplikacja_bazodanowa)

### 4. Testy jednostkowe

Testy jednostkowe istnieją w postaci dwóch plików w głównym folderze projektu: [test\\_database.py](#) oraz [test\\_mainwindow.py](#).

Aby je uruchomić, należy posiadać zainstalowane narzędzie [pytest](#) i uruchomić je w folderze projektu podając nazwę pliku, na przykład `pytest test_database.py`.

### 5. Dokumentacja kodu

Dokumentacja kodu została wygenerowana za pomocą narzędzia [pdoc](#) i jest dostępna jako pliki HTML w folderze [docs](#) w kodzie źródłowym projektu.

### 6. Instrukcja instalacji i uruchomienia aplikacji

- Pobranie gotowych plików wykonywalnych
  1. Przejdź na [tę stronę](#) i pobierz plik klikając [Database\\_Application.zip](#) w sekcji Assets
  2. Rozpakuj plik za pomocą oprogramowania typu WinRAR, 7-Zip
  3. Uruchom plik [Database Application](#) znajdujący się w folderze
- Uruchomienie z kodu źródłowego
  1. Upewnij się, że masz zainstalowane: Python 3.12 (lub nowszy), git
  2. Sklonuj repozytorium za pomocą narzędzia git: `git clone https://github.com/PiotrKurczab/aplikacja_bazodanowa.git`
  3. Przejdź do folderu `aplikacja_bazodanowa`
  4. Wykonaj komendę `pip install -r requirements.txt`
  5. Uruchom aplikację za pomocą komendy `py main.py`

## 7. Instrukcja użytkowania aplikacji

1. Po uruchomieniu aplikacji wyświetlone zostanie okno główne z zakładkami dla różnych tabel bazy danych: Customers, Orders, Products, Suppliers oraz zakładka Customer Orders, która łączy dane z tabel Customers, Orders i Products.
2. W zakładkach widoczna jest tabela z odpowiednimi danymi, pole wyszukiwania i przycisk filtrowania.
3. Aby dodać nowy rekord, kliknij przycisk "Add" i wypełnij formularz z odpowiednimi danymi.
4. Aby usunąć rekord, zaznacz go w tabeli, a następnie kliknij przycisk "Delete".
5. Aby edytować wartość w rekordzie, kliknij dwukrotnie na odpowiednią komórkę w tabeli i wprowadź nową wartość.
6. Aby wyszukać rekordy, wpisz frazę w polu wyszukiwania. Wyszukiwanie jest wykonywane po każdej zmianie tekstu w polu.
7. Aby filtrować dane, kliknij przycisk "Filter", zaznacz odpowiednie kryteria filtrowania w wyświetlonym oknie oraz wypełnij swoje kryteria.
8. Aby wyeksportować dane bazy danych do pliku CSV, kliknij przycisk "Export" i wybierz lokalizację oraz nazwę pliku.
9. Aby zaimportować dane z pliku CSV, kliknij przycisk "Import" i wybierz plik CSV z danymi.
10. Zakładka "Customer Orders" łączy dane z tabel Customers, Orders i Products, wyświetlając listę zamówień wraz z informacjami o kliencie i produkcie.
11. Po wprowadzeniu zmian w danych, zakładka "Zamówienia klientów" jest automatycznie odświeżana, aby odzwierciedlić najnowsze dane.
12. Aby zamknąć aplikację, zamknij główne okno programu.