

# Bazy danych

Kierunek

*Informatyczne Systemy Automatyki*

Termin

*środa TN 18<sup>55</sup> – 20<sup>35</sup>*

Imię, nazwisko, numer albumu

*Igor Frysiak 272548*

Link do projektu

[https://github.com/IFrysiak/Bazy\\_danych](https://github.com/IFrysiak/Bazy_danych)



## ETAP 1

---

### Contents

1	Cel etapu	1
2	Temat projektu	1
3	Skład zespołu	1
4	Zakres projektu	2
4.1	User story . . . . .	2
4.2	Wymagania . . . . .	2
4.3	Przypadki użycia . . . . .	2
4.4	Diagram . . . . .	3
5	Wybrane technologie	4
6	Projekt bazy danych	4
7	Możliwe transakcje	4
8	Prognoza charakteru poszczególnych encji	5
9	Zmiany	8
10	Implementacja backend	8

## 1 Cel etapu

Celem tego etapu jest określenie kolejno:

- tematu projektu
- składu zespołu
- zakresu projektu

## 2 Temat projektu

Celem projektu jest stworzenie systemu biblioteki korzystającego z baz danych klientów, książek oraz wypożyczeń.

## 3 Skład zespołu

Igor Frysiak

nr albumu: 272548

## 4 Zakres projektu

### 4.1 User story

Przykładowe historyjki użytkowników:

- Jako czytelnik chciałbym móc w prosty sposób sprawdzić czy dana książka jest dostępna w bibliotece bez chodzenia i szukania jej.
- Jako czytelnik chciałbym mieć podgląd wypożyczonych przeze mnie książek oraz terminu ich oddania.
- Jako czytelniczka chciałabym szybko wypożyczać książki oraz przedłużać terminy oddawania bez kontaktu z pracownikami.
- Jako czytelnik chciałbym mieć dostęp do swoich danych podanych w czasie rejestracji oraz mieć możliwość ich edytowania.
- Jako pracownik chciałabym mieć dostęp do spisu książek, klientów oraz dostęp do informacji o wypożyczeniach.

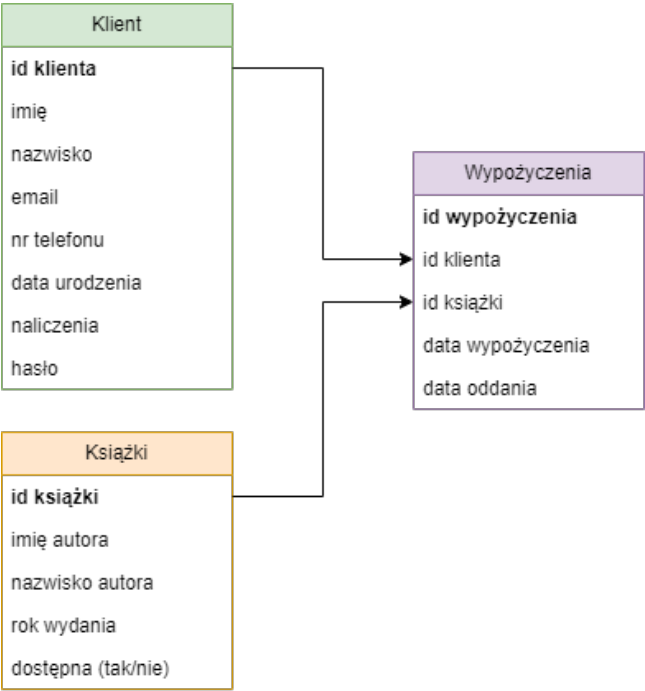
### 4.2 Wymagania

- Użytkownik otwierający stronę ma wypisany spis książek znajdujących się w bibliotece (dostępnych oraz tymczasowo niedostępnych).
- Użytkownik zainteresowany wypożyczeniem musi założyć swoje konto, jeśli takiego nie posiada.
- Podczas logowania stwierdzane jest kto próbuje się zalogować: czytelnik czy pracownik biblioteki.
- Zalogowany klient ma dostęp do:
  - swoich danych podanych w czasie rejestracji
  - edycji swoich danych
  - usunięcia swojego konta
  - sprawdzenia czy ma zaległości do zapłacenia
  - wypożyczenia książek
- Zalogowany pracownik ma dostęp do:
  - podejrzenia danych klientów (bez hasła)
  - dodawania/usuwania książek ze spisu
  - spisu wypożyczeń
  - usunięcia konta klienta

### 4.3 Przypadki użycia

Taki system sprawdziłby się na przykład w kiosku w bibliotece publicznej. Czytelnicy mogliby w łatwy sposób dokonywać rejestracji oraz wypożyczać książki bez pomocy pracownika. Zwolniło by to pracownika z niektórych obowiązków, takich jak: rejestrowanie nowych czytelników, sprawdzanie dostępności książek, zmiana danych klientów, przekazywanie informacji o zaległościach.

4.4 Diagram

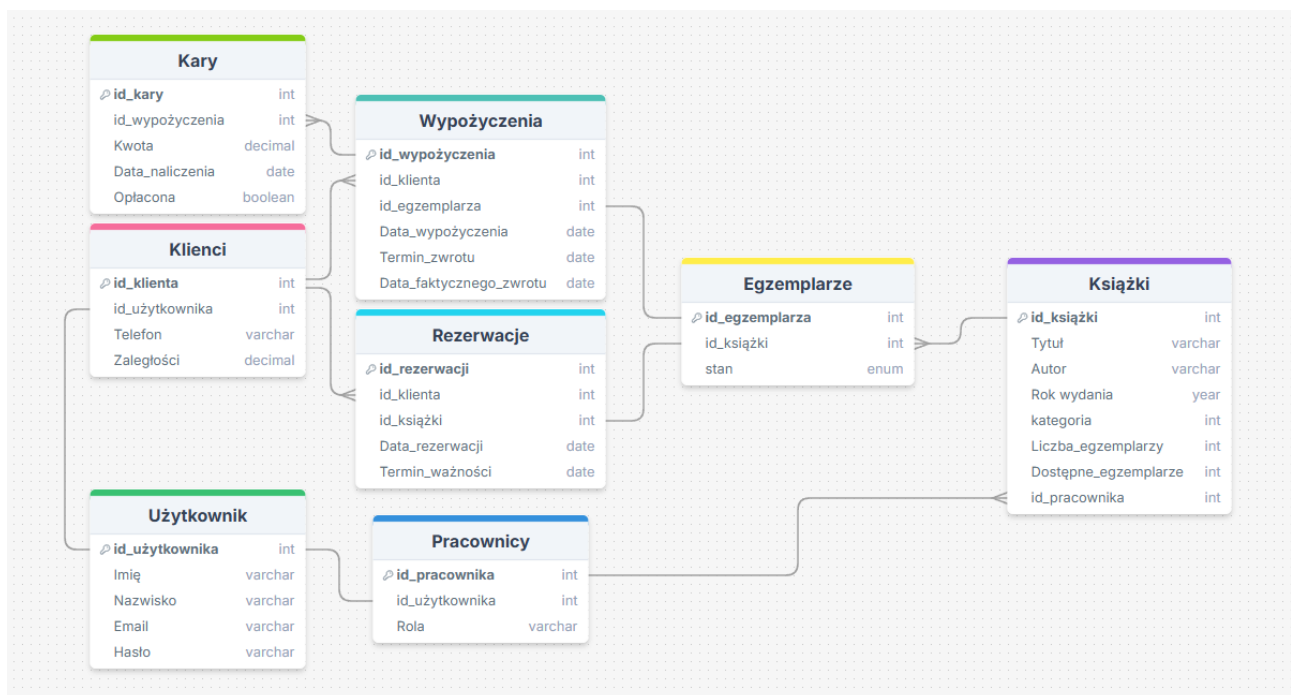


Wstępny diagram

## 5 Wybrane technologie

- Aplikacja desktopowa - użytkownicy mogą na dowolnym urządzeniu korzystać z systemu
- Baza danych MySQL
- Technologia backend - Flask, python
- Technologia front end - React, JavaScript

## 6 Projekt bazy danych



Baza danych biblioteki

## 7 Możliwe transakcje

- Rejestracja nowego klienta
  - Dodanie nowego klienta do tabeli Klienci.
  - Powiązanie nowego użytkownika z tabelą Użytkownik.
- Zmiana danych klienta przez klienta
  - Aktualizacja danych w tabeli Klienci i Użytkownik.
- Usunięcie klienta przez pracownika
  - Sprawdzenie, czy klient nie ma zaległości w wypożyczeniach.
  - Usunięcie danych klienta z tabeli Klienci.
- Dodanie nowej książki przez pracownika
  - Dodanie książki do tabeli Książki.
  - Powiązanie książki z pracownikiem.
  - Określenie liczby dostępnych egzemplarzy.

- Dodanie nowego egzemplarza książki
  - Dodanie egzemplarza do tabeli Egzemplarze.
- Klient wypożycza książkę z biblioteki
  - Utworzenie nowego wpisu w tabeli Wypożyczenia.
  - Zmniejszenie liczby dostępnych egzemplarzy w tabeli Książki.
  - Powiązanie wypożyczenia z klientem i egzemplarzem.
- Zwrot książki
  - Aktualizacja daty zwrotu w tabeli Wypożyczenia.
  - Zwiększenie liczby dostępnych egzemplarzy w tabeli Książki.
  - Zmiana stanu egzemplarza w tabeli Egzemplarze.
  - Sprawdzenie, czy książka jest spóźniona i ewentualne naliczenie kary.
- Rezerwacja książki
  - Utworzenie nowego wpisu w tabeli Rezerwacje.
  - Powiązanie rezerwacji z konkretnym egzemplarzem.
  - Zmniejszenie liczby dostępnych egzemplarzy w tabeli Książki.
- Naliczanie kary za opóźnienie
  - Sprawdzenie, czy termin zwrotu książki został przekroczony.
  - Dodanie wpisu do tabeli Kary.

## 8 Prognoza charakteru poszczególnych encji

### 1. Encja: Klienci

- **Przewidywany sposób użycia:**
  - **Odczyt:** Klienci będą często odczytywani, gdy pracownicy lub system będą sprawdzać dane klienta (np. przy wypożyczeniu książki, rezerwacji, przeglądaniu historii wypożyczeń).
  - **Zapis:** Rzadziej, zazwyczaj podczas rejestracji nowych użytkowników (nowych klientów).
- **Przewidywana zmienność:**
  - **Dodawanie:** Klienci będą dodawani stosunkowo rzadko, w miarę jak nowi użytkownicy będą się rejestrować.
  - **Edycja:** Zmienność w edytowaniu danych klienta będzie niewielka. Zmiany mogą obejmować dane kontaktowe (np. adres, telefon), ale te zmiany będą miały miejsce stosunkowo rzadko.
  - **Usuwanie:** Usuwanie klientów będzie miało miejsce głównie w wyjątkowych sytuacjach, na życzenie klienta.
- **Przewidywana liczba wystąpień obiektów:** Zależna od wielkości biblioteki. Możemy przewidywać od kilkuset do kilku tysięcy klientów, w zależności od wielkości bazy użytkowników.

### 2. Encja: Książki

- **Przewidywany sposób użycia:**
  - **Odczyt:** Bardzo często, ponieważ książki będą przeglądane przez pracowników i klientów (w celu sprawdzenia dostępności, rezerwacji, wypożyczenia).
  - **Zapis:** Książki będą dodawane rzadziej, głównie przez pracowników, gdy pojawią się nowe tytuły.
- **Przewidywana zmienność:**
  - **Dodawanie:** Książki będą dodawane do bazy sporadycznie, zwykle wtedy, gdy pojawią się nowe pozycje w bibliotece.
  - **Edycja:** Zmiany będą dotyczyć głównie liczby dostępnych egzemplarzy.
  - **Usuwanie:** Raczej nie przewidziane.
- **Przewidywana liczba wystąpień obiektów:** W zależności od wielkości biblioteki, może to być od kilkuset do kilku tysięcy książek w systemie.

### 3. Encja: Egzemplarze

- **Przewidywany sposób użycia:**

- **Odczyt:** Dane o egzemplarzach będą często odczytywane w celu sprawdzenia ich dostępności, stanu lub przy realizacji wypożyczeń i rezerwacji książek.
- **Zapis:** Zapis będzie wykonywany przy dodawaniu nowych egzemplarzy do systemu lub aktualizacji ich stanu (np. zmiana na "uszkodzony" lub "zarezerwowany").

- **Przewidywana zmienność:**

- **Dodawanie:** Nowe egzemplarze będą regularnie dodawane przy zakupie nowych książek do biblioteki lub przy uzupełnianiu brakujących egzemplarzy.
- **Edycja:** Zmiany w danych egzemplarzy będą sporadyczne i głównie dotyczyć ich stanu (np. oznaczenie jako "uszkodzony").
- **Usuwanie:** Zmiana stanu na "usunięty".

- **Przewidywana liczba wystąpień obiektów:** Liczba egzemplarzy będzie różna w zależności od wielkości biblioteki, ale w dużych bibliotekach może wynosić od kilku do kilkudziesięciu tysięcy.

### 4. Encja: Pracownicy

- **Przewidywany sposób użycia:**

- **Odczyt:** Pracownicy będą rzadko odczytywani, głównie w kontekście logowania, przypisywania książek lub monitorowania działań.
- **Zapis:** Pracownicy będą dodawani głównie w procesie rekrutacji lub przy zmianach w zespole.

- **Przewidywana zmienność:**

- **Dodawanie:** Dodawanie nowych pracowników będzie miało miejsce sporadycznie, zazwyczaj gdy nowi pracownicy są zatrudniani.
- **Edycja:** Zmiany mogą dotyczyć danych kontaktowych lub ról w systemie (np. zmiana stanowiska), ale zmiany będą sporadyczne.
- **Usuwanie:** Pracownicy będą usuwani w przypadku zakończenia ich pracy, co może zdarzyć się stosunkowo rzadko.

- **Przewidywana liczba wystąpień obiektów:** Liczba pracowników będzie raczej niewielka, od kilku do kilkudziesięciu, w zależności od wielkości biblioteki.

### 5. Encja: Wypożyczenia

- **Przewidywany sposób użycia:**

- **Odczyt:** Sprawdzanie statusu wypożyczeń.
- **Zapis:** Tworzenie nowych wpisów w momencie wypożyczeń.

### 6. Encja: Rezerwacje

- **Przewidywany sposób użycia:**

- **Odczyt:** Często odczytywane w celu sprawdzenia statusu rezerwacji przez pracowników i klientów.
- **Zapis:** Zapis jest wykonywany, gdy klient dokonuje rezerwacji książki, a książka jest niedostępna.

- **Przewidywana zmienność:**

- **Dodawanie:** Rezerwacje będą dodawane do systemu w miarę potrzeb.
- **Edycja:** Zmiany w rezerwacjach mogą wystąpić przy anulowaniu rezerwacji, ale są rzadkie.
- **Usuwanie:** Usuwanie będzie miało miejsce, gdy rezerwacja wygasa lub książka zostanie odebrana przez klienta.

- **Przewidywana liczba wystąpień obiektów:** Liczba rezerwacji zależy od liczby dostępnych książek i liczby klientów, ale może wynosić nawet kilkaset w roku.

## 7. Encja: Kary

- **Przewidywany sposób użycia:**
  - **Odczyt:** Odczytywane, aby sprawdzić stan zaległości klientów.
  - **Zapis:** Zapis następuje, gdy naliczana jest kara za opóźniony zwrot książki.
- **Przewidywana zmienność:**
  - **Dodawanie:** Kary będą dodawane regularnie, jeśli klienci nie oddadzą książek na czas.
  - **Edycja:** Zmiany następują, gdy klient opłaci karę.
  - **Usuwanie:** Usuwanie będzie miało miejsce, gdy kara zostanie opłacona.
- **Przewidywana liczba wystąpień obiektów:** Zwykle liczba kar będzie zależna od liczby spóźnionych zwrotów, ale przewidywana liczba to kilka tysięcy rocznie w dużych bibliotekach.

## 8. Encja: Użytkownik

- **Przewidywany sposób użycia:**
  - **Odczyt:** Dane użytkownika będą często odczytywane w celu sprawdzania historii wypożyczeń, rezerwacji i stanu konta.
  - **Zapis:** Zapis następuje podczas rejestracji nowego użytkownika lub aktualizacji jego danych (np. adresu e-mail, numeru telefonu).
- **Przewidywana zmienność:**
  - **Dodawanie:** Nowi użytkownicy będą regularnie dodawani, szczególnie na początku roku akademickiego lub w okresach promocji biblioteki.
  - **Edycja:** Edycje danych użytkownika będą sporadyczne, związane z aktualizacjami danych kontaktowych lub zmianami statusu konta.
  - **Usuwanie:** Usuwanie użytkowników może wystąpić w przypadku nieaktywności przez dłuższy czas lub na prośbę użytkownika.
- **Przewidywana liczba wystąpień obiektów:** Liczba użytkowników będzie się różnić w zależności od rozmiaru biblioteki, ale w dużych bibliotekach może wynosić kilkanaście tysięcy aktywnych kont.

## 9 Zmiany

- Do implementacji bazy danych wykorzystano SQLite, ze względu na brak potrzeby instalacji (jest on domyślnie zainstalowany w pythonie) oraz na obecnym etapie projektu ma on wystarczającą funkcjonalność. W przyszłości, gdy baza danych zasilona będzie większą ilością informacji, możliwa jest potencjalna zmiana na MySQL. Taka zmiana nie będzie się natomiast wiązała z dużymi modyfikacjami w kodzie.
- Zmiana z zadeklarowanej aplikacji dekstopowej na webową, dzięki tej zmianie korzystanie z aplikacji będzie możliwe na każdym urządzeniu z dostępem do internetu.
- W tabeli **Kary** planowane było aby każdy dzień spóźnienia z oddaniem książki zapisywany był oddzielnie, zmieniono to podejście na takie wymagające mniej wpisów w tabeli, zapewni to zmniejszenie wielkości bazy danych i szybsze jej działanie. Każdego dnia kwota będzie zwiększana do czasu opłacenia zaległości. *Data\_naliczenia* została zmieniona na *Data\_ostatniego\_naliczenia*, która będzie datą ostatniego naliczenia przed opłaceniem zaległości.

## 10 Implementacja backend

Symulacja zapytań pochodzących z frontendu została zaimplementowana z użyciem narzędzia Postman. Przewidziane operacje na tabelach to:

- Użytkownicy (USER)
  - **GET ALL USERS** - operacja GET  
Wyświetla wszystkich użytkowników systemu.
  - **ADD USER** - operacja POST  
Dodaje nowego użytkownika. Przykładowy kod JSON:

```
{
  "imie": "Jan",
  "nazwisko": "Kowalski",
  "email": "janek@gmail.com",
  "haslo": "haslo123"
}
```
  - **UPDATE USER** - operacja PATCH  
Uaktualnia dane użytkownika o podanym id.
  - **DELETE USER** - operacja DELETE  
Usuwa użytkownika.
- Klienci (CUSTOMERS)
  - **GET ALL CUSTOMERS** - operacja GET  
Wyświetla wszystkich klientów biblioteki.
  - **GET CUSTOMER** - operacja GET  
Wyświetla informacje o kliencie o podanym id.
  - **ADD CUSTOMER** - operacja POST  
Dodaje nowego klienta. Przykładowy kod JSON:

```
{
  "id_uzytkownika": "1",
  "telefon": "123456789"
}
```
  - **UPDATE CUSTOMER** - operacja PATCH  
Uaktualnia dane klienta o podanym id.
  - **DELETE CUSTOMER** - operacja DELETE  
Usuwa klienta. Gdy ma zaległości nie pozwala usunąć konta.



- Pracownicy (EMPLOYEE)

- **GET ALL EMPLOYEES** - operacja GET  
Wyświetla wszystkich pracowników biblioteki.
- **ADD EMPLOYEE** - operacja POST  
Dodaje nowego pracownika. Przykładowy kod JSON:

```
{  "id_uzytkownika": "1",  "rola": "szef"}
```
- **UPDATE EMPLOYEE** - operacja PATCH  
Uaktualnia dane pracownika o podanym id.
- **DELETE EMPLOYEE** - operacja DELETE  
Usuwa pracownika.

- Książki (BOOK)

- **GET ALL BOOKS** - operacja typu GET  
Wyświetla wszystkie książki w bibliotece.
- **GET BOOK** - operacja GET  
Wyświetla informacje o książce o podanym id.
- **ADD BOOK** - operacja POST  
Dodaje książkę. Przykładowy kod JSON:

```
{  "tytul": "Pan Tadeusz",  "autor": "Adam Mickiewicz",  "rok_wydania": "1834",  "kategoria": "poezja epicka",  "id_pracownika": "1"}
```
- **DELETE BOOK** - operacja DELETE  
Usuwa książkę.
- **UPDATE BOOK** - operacja PATCH  
Dodaje możliwość modyfikacji informacji o książce o podanym id.

- Egzemplarze książek (COPY OF BOOKS)

- **GET ALL COPIES** - operacja GET  
Wyświetla wszystkie egzemplarze książek w systemie.
- **GET ALL COPIES OF BOOK** - operacja GET  
Wyświetla wszystkie egzemplarze danej książki.
- **ADD COPY** - operacja POST  
Dodaje nowy egzemplarz książki. Przykładowy kod JSON:

```
{  "id_ksiazki": "1",  "stan": "dostepna"}
```

Możliwe stany książek to: "dostępna", "wypożyczona", "zarezerwowana", "zniszczona".
- **UPDATE COPY** - operacja PATCH  
Uaktualnia informacje o egzemplarzu książki.
- **DELETE COPY** - operacja DELETE  
Usuwa egzemplarz książki.

- Wypożyczenia książek (BOOK LOANS)

- **GET ALL LOANS** - operacja GET  
Wyświetla wszystkie wypożyczenia książek.
- **ADD LOAN** - operacja POST  
Dodaje nowe wypożyczenie. Przykładowy kod JSON:
 

```
{
  "id_klienta": "1",
  "id_egzemplarza": "1",
  "data_wypozyczenia": "2024-12-28",
  "termin_zwrotu": "2024-12-29"
}
```
- **UPDATE LOAN/RETURN** - operacja PATCH  
Aktualizuje szczegóły wypożyczenia lub oznacza książkę jako zwróconą nadając jej *data-faktycznego-zwrotu*.
- **DELETE LOAN** - operacja DELETE  
Usuwa wypożyczenie.
- Rezerwacje (RESERVATIONS)
  - **GET ALL RESERVATIONS** - operacja GET  
Wyświetla wszystkie rezerwacje w systemie.
  - **ADD RESERVATION** - operacja POST  
Dodaje nową rezerwację. Przykładowy kod JSON:
 

```
{
    "id_klienta": "1",
    "id_ksiazki": "1",
    "data_rezerwacji": "2024-12-30",
    "termin_waznosci": "2024-12-31"
  }
```
  - **UPDATE RESERVATION** - operacja PATCH  
Uaktualnia szczegóły rezerwacji.
  - **DELETE RESERVATION** - operacja DELETE  
Usuwa rezerwację.
- Kary (FINES)
  - **GET ALL FINES** - operacja GET  
Wyświetla wszystkie kary.
  - **ADD FINE** - operacja POST  
Dodaje nową karę. Przykładowy kod JSON:
 

```
{
    "id_wypozyczenia": "1",
    "kwota": "2",
    "data_ostatniego_naliczenia": "2024-12-29",
    "oplacona": false
  }
```
  - **UPDATE FINE** - operacja PATCH  
Uaktualnia szczegóły nałożonej kary.
  - **DELETE FINE** - operacja DELETE  
Usuwa karę.