

Jan Lewandowski,    Szymon Jończak-Lis,  
Cezary Czetyrba,    Michał Mickiewicz

# Dokumentacja projektu z baz danych

20.01.2025

## 1. Spis użytych technologii

- Python
  - wersja  $\geq 3.9$
  - moduły:
    - pandas
    - mysql-connector-python
    - python-dotenv
- R
  - wersja  $\geq 4.4.2$
  - pakiety:
    - RMariaDB
    - ggplot2
    - ggrepel
    - lattice
    - dplyr
    - tidyverse
    - ggpubr
- RStudio
  - wersja  $\geq 2024.12.0+467$
- Visual Studio Code
  - Dodatki
    - ERD editor
    - Inline SQL
- Git, GitHub
- Git Bash w systemie Windows

## 2. Struktura projektu

```
/
├── .github/
├── dane_statystyczne/
│   ├── Imiona_nadane_dzieciom_w_Polsce_w_*.csv
│   ├── nazwiska_meskie.csv
│   └── nazwiska_zenskie.csv
├── dokumentacja/
│   ├── dokumentacja.pdf
│   ├── dokumentacja.tex
│   └── schemat_bazy_danych.vuerd
├── obrazy/
│   ├── baza_danych.fullsize.png
│   └── baza_danych.png
├── raport/
│   ├── raport.html
│   ├── raport.pdf
│   └── raport.Rmd
├── wypelnienie/
│   ├── custom_util.py
│   ├── do_wypelnienia.py
│   ├── sql.py
│   └── wypelnienie.py
├── .env.example
├── .gitignore
├── LICENSE
├── README.md
└── requirements.txt
```

Projekt podzielony jest na główne foldery i pliki:

- **.github**
  - Folder zawierający pliki konfiguracyjne dla GitHub Actions. W naszym przypadku jest to analiza stylistyczna kodu.
- **dane\_statystyczne**
  - Folder zawierający pliki z danymi statystycznymi. Są to pliki CSV z imionami i nazwiskami.
    - Źródła:
      - [Nazwiska w rejestrze PESEL](#)
      - [Imiona nadane dzieciom w Polsce](#)
- **dokumentacja**
  - Folder zawierający pliki z dokumentacją projektu. Znajduje się tam plik `dokumentacja.pdf`, który jest wygenerowany z pliku `dokumentacja.tex` oraz `schemat_bazy_danych.vuerd` zawierający schemat bazy danych w formacie ERD Editor.
- **obrazy**

- Folder zawierający obrazy wykorzystane w raporcie oraz dokumentacji.
- **raport**
  - Folder zawierający pliki z raportem projektu. Znajdują się tam pliki `raport.pdf` oraz `raport.html`, które są generowane z pliku `raport.Rmd`.
- **wypelnienie**
  - Folder zawierający skrypty do wypełnienia bazy danych. Znajduje się tam główny plik `wypelnienie.py`, który wypełnia bazę danych.
    - `custom_util.py` - plik z funkcjami pomocniczymi.
    - `do_wypelnienia.py` - plik z danymi na temat wycieczek i pracowników.
    - `sql.py` - plik z zapytaniami SQL.
- **.env.example**
  - Przykładowy plik środowiskowy.
- **.gitignore**
  - Lista plików ignorowanych przez Git.
- **LICENSE**
  - Licencja projektu.
- **README.md**
  - Ogólny opis projektu.

### 3. Instrukcja uruchomienia

Aby uruchomić projekt potrzebny jest Python w wersji 3.9 lub nowszej. Po sklonowaniu repozytorium należy zainstalować wymagane moduły Pythona:

```
1 cd projekt-bazy-danych
2 pip install -r requirements.txt
```

Następnie należy skopiować plik `.env.example` i zmienić jego nazwę na `.env`. W pliku tym należy podać hasło do połączenia z bazą danych.

```
1 cp .env.example .env
2 vim .env
```

Po zainstalowaniu modułów i uzupełnieniu pliku `.env` można uruchomić skrypt wypełniający bazę danych.

```
1 python wypelnienie/wypelnienie.py
```

Kiedy zakończy się on poprawnie wypisując Wypelniono baze danych, możemy wygenerować raport. Można go otworzyć w RStudio i uruchomić przyciskiem Knit lub mając Rscript i pandoc poprzez umieszczenie



## 5. Analiza tabel

### 5.1. Tabela `rodzaje_transportu`

Tabela jest postaci:

`rodzaje_transportu(id_rodzaju_transportu, nazwa, opis)`

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_rodzaju\_transportu \rightarrow nazwa$$

$$id\_rodzaju\_transportu \rightarrow opis$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.2. Tabela `koszty_miasta`

Tabela jest postaci:

`koszty_miasta(id_kosztu_miasta, koszt, cena_dla_klienta, id_miasta)`

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_kosztu\_miasta \rightarrow koszt$$

$$id\_kosztu\_miasta \rightarrow cena\_dla\_klienta$$

$$id\_kosztu\_miasta \rightarrow id\_miasta$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.3. Tabela `miasta`

Tabela jest postaci:

`miasta(id_miasta, miasto, id_kraju)`

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_miasta \rightarrow miasto$$

$$id\_miasta \rightarrow id\_kraju$$

Zakładamy, że miasto nie definiuje jednoznacznie kraju. Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.4. Tabela `kraje`

Tabela jest postaci:

`kraje(id_kraju, kraj)`

Jedyną nietrywialną zależnością funkcyjną jest:

$$id\_kraju \rightarrow kraj$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.5. Tabela transport\_propozycja\_wycieczki

Tabela jest postaci:

transport\_propozycja\_wycieczki(id\_rodzaju\_transportu, id\_propozycji, id\_kosztu\_miasta)

W tej tabeli nie występują nietrywialne zależności funkcyjne, wszystkie trzy kolumny tworzą klucz elementarny, a pomiędzy nimi nie ma zależności. Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.6. Tabela adresy

Tabela jest postaci:

adresy(id\_adresu, adres, adres2, id\_miasta, kod\_pocztowy)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_adresu \rightarrow adres$$

$$id\_adresu \rightarrow adres2$$

$$id\_adresu \rightarrow id\_miasta$$

$$id\_adresu \rightarrow kod\_pocztowy$$

Zakładamy, że kod pocztowy nie definiuje jednoznacznie miasta i adres nie definiuje kodu pocztowego.

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.7. Tabela miejsca\_wycieczki

Tabela jest postaci:

miejsca\_wycieczki(id\_miejsca, nazwa, id\_adresu, koszt, cena\_dla\_klienta, id\_kontrahenta)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_miejsca \rightarrow nazwa$$

$$id\_miejsca \rightarrow id\_adresu$$

$$id\_miejsca \rightarrow koszt$$

$$id\_miejsca \rightarrow cena\_dla\_klienta$$

$$id\_miejsca \rightarrow id\_kontrahenta$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.8. Tabela propozycja\_koszt\_u\_kontrahenta

Tabela jest postaci:

propozycja\_koszt\_u\_kontrahenta(id\_propozycji, id\_kosztu\_u\_kontrahenta)

W tej tabeli nie występują nietrywialne zależności funkcyjne, wszystkie trzy kolumny tworzą klucz elementarny, a pomiędzy nimi nie ma zależności. Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.9. Tabela koszty\_u\_kontrahenta

Tabela jest postaci:

koszty\_u\_kontrahenta(id\_kosztu\_u\_kontrahenta, nazwa, koszt, cena\_dla\_klienta, id\_kontrahenta)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_kosztu\_u\_kontrahenta \rightarrow nazwa$$

$$id\_kosztu\_u\_kontrahenta \rightarrow koszt$$

$$id\_kosztu\_u\_kontrahenta \rightarrow cena\_dla\_klienta$$

$$id\_kosztu\_u\_kontrahenta \rightarrow id\_kontrahenta$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.10. Tabela kontrahenci

Tabela jest postaci:

kontrahenci(id\_kontrahenta, nazwa, opis, email, id\_adresu)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_kontrahenta \rightarrow nazwa$$

$$id\_kontrahenta \rightarrow opis$$

$$id\_kontrahenta \rightarrow email$$

$$id\_kontrahenta \rightarrow id\_adresu$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.11. Tabela propozycje\_wycieczki

Tabela jest postaci:

propozycje\_wycieczki(id\_propozycji, nazwa, opis, ograniczenia, min\_liczba\_osob, max\_liczba\_osob, nasze\_koszty\_razem, cena\_dla\_klienta, id\_miejsca)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_propozycji \rightarrow nazwa$$

$$id\_propozycji \rightarrow opis$$

$$id\_propozycji \rightarrow ograniczenia$$

$$id\_propozycji \rightarrow min\_liczba\_osob$$

$$id\_propozycji \rightarrow max\_liczba\_osob$$

$$id\_propozycji \rightarrow nasze\_koszty\_razem$$

$$id\_propozycji \rightarrow cena\_dla\_klienta$$

$$id\_propozycji \rightarrow id\_miejsca$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.12. Tabela wycieczki

Tabela jest postaci:

wycieczki(id\_wycieczki, czas\_wyjazdu, czas\_powrotu, liczba\_osob, id\_propozycji)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_wycieczki \rightarrow czas\_wyjazdu$$

$$id\_wycieczki \rightarrow czas\_powrotu$$

$$id\_wycieczki \rightarrow liczba\_osob$$

$$id\_wycieczki \rightarrow id\_propozycji$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.13. Tabela transakcje\_kontrahenci

Tabela jest postaci:

transakcje\_kontrahenci(id\_transakcji, kwota, data\_transakcji, id\_kontrahenta, id\_wycieczki)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_transakcji \rightarrow kwota$$

$$id\_transakcji \rightarrow czas\_wyjazdu$$

$$id\_transakcji \rightarrow data\_transakcji$$

$$id\_transakcji \rightarrow id\_kontrahenta$$

$$id\_transakcji \rightarrow id\_wycieczki$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.14. Tabela rodzaje\_uslug\_dodatkowych

Tabela jest postaci:

rodzaje\_uslug\_dodatkowych(id\_uslugi\_dodatkowej, nazwa, opis\_uslugi, koszt, cena\_dla\_klienta, id\_kontrahenta)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_uslugi\_dodatkowej \rightarrow nazwa$$

$$id\_uslugi\_dodatkowej \rightarrow opis\_uslugi$$

$$id\_uslugi\_dodatkowej \rightarrow koszt$$

$$id\_uslugi\_dodatkowej \rightarrow cena\_dla\_klienta$$

$$id\_uslugi\_dodatkowej \rightarrow id\_kontrahenta$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.



### 5.15. Tabela uslugi\_dodatkowe

Tabela jest postaci:

uslugi\_dodatkowe(id\_wycieczki, id\_uslugi)

W tej tabeli nie występują nietrywialne zależności funkcyjne, dwie kolumny tworzą klucz elementarny, a pomiędzy nimi nie ma zależności. Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.16. Tabela pracownik\_wycieczka

Tabela jest postaci:

pracownik\_wycieczka(id\_wycieczki, id\_pracownika)

W tej tabeli nie występują nietrywialne zależności funkcyjne, dwie kolumny tworzą klucz elementarny, a pomiędzy nimi nie ma zależności. Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.17. Tabela pracownicy

Tabela jest postaci:

pracownicy(id\_pracownika, imie, nazwisko, id\_stanowiska, id\_telefonu, uwagi)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_pracownika \rightarrow imie$$

$$id\_pracownika \rightarrow nazwisko$$

$$id\_pracownika \rightarrow id\_stanowiska$$

$$id\_pracownika \rightarrow id\_telefonu$$

$$id\_pracownika \rightarrow uwagi$$

Zakładamy, że imię i nazwisko nie identyfikują jednoznacznie pracownika. Skoro każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.18. Tabela telefony

Tabela jest postaci:

telefony(id\_telefonu, telefon, numer\_bliskiego)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_telefonu \rightarrow telefon$$

$$id\_telefonu \rightarrow numer\_bliskiego$$

Zakładamy, że numery telefonów nie są unikalne, zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.19. Tabela stanowiska

Tabela jest postaci:

stanowiska(id\_stanowiska, nazwa\_stanowiska, wysokosc\_pensji)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_stanowiska \rightarrow nazwa\_stanowiska$$

$$id\_stanowiska \rightarrow wysokosc\_pensji$$

Zakładamy, że nie ma zależności, które zaczynają się od stanowiska, zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.20. Tabela klient\_wycieczka

Tabela jest postaci:

klient\_wycieczka(id\_wycieczki, id\_klienta)

W tej tabeli nie występują nietrywialne zależności funkcyjne, dwie kolumny tworzą klucz elementarny, a pomiędzy nimi nie ma zależności. Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.21. Tabela transakcje\_pracownicy

Tabela jest postaci:

transakcje\_pracownicy(id\_transakcji, kwota, data\_transakcji, id\_pracownik)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_transakcji \rightarrow kwota$$

$$id\_transakcji \rightarrow czas\_wyjazdu$$

$$id\_transakcji \rightarrow data\_transakcji$$

$$id\_transakcji \rightarrow id\_pracownika$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.22. Tabela transakcje\_klienci

Tabela jest postaci:

transakcje\_klienci(id\_transakcji, kwota, data\_transakcji, id\_klienta, id\_wycieczki)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_transakcji \rightarrow kwota$$

$$id\_transakcji \rightarrow czas\_wyjazdu$$

$$id\_transakcji \rightarrow data\_transakcji$$

$$id\_transakcji \rightarrow id\_klienta$$

$$id\_transakcji \rightarrow id\_wycieczki$$

Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

### 5.23. Tabela klienci

Tabela jest postaci:

klienci(id\_klienta, imie, nazwisko, plec, email, id\_telefonu)

Jedynymi nietrywialnymi zależnościami funkcyjnymi są:

$$id\_klienta \rightarrow imie$$

$$id\_klienta \rightarrow nazwisko$$

$$id\_klienta \rightarrow plec$$

$$id\_klienta \rightarrow email$$

$$id\_klienta \rightarrow id\_telefonu$$

Zakładamy, że imię i nazwisko nie identyfikują jednoznacznie klienta. Zatem każda nietrywialna zależność funkcyjna zaczyna się od nadklucza i tabela spełnia wymagania bycia w EKNF.

W żadnym połączeniu tabel nie ma zależności tranzytywnych, każda tabela pojedynczo jest w EKNF, zatem to pokazuje, że cała baza danych jest w EKNF.

## 6. Co było najtrudniejsze w realizacji projektu

- Dopasowanie realistycznych kosztów usług i podróży.
- Aktualizacja wypełniania bazy danych po zmianach w schemacie.