

Dokumentacja Projektu Zaliczeniowego  
Baza danych dla firmy Wombat Grylls sp. z o.o.

Kacper Daniel, Tadeusz Jagniewski, Paweł Karwecki, Piotr Marciniak, Konrad Mądry

20 stycznia 2025

## Spis treści

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Spis użytych technologii</b>                        | <b>3</b> |
| <b>2</b> | <b>Lista plików i opis ich zawartości</b>              | <b>3</b> |
| <b>3</b> | <b>Kolejność i sposób uruchamiania plików</b>          | <b>3</b> |
| <b>4</b> | <b>Schemat projektu bazy danych</b>                    | <b>3</b> |
| <b>5</b> | <b>Lista zależności funkcyjnych dla każdej relacji</b> | <b>4</b> |
| 5.1      | Tabela: Address . . . . .                              | 4        |
| 5.2      | Tabela: Costs . . . . .                                | 4        |
| 5.3      | Tabela: Customers . . . . .                            | 4        |
| 5.4      | Tabela: Payment . . . . .                              | 4        |
| 5.5      | Tabela: Staff . . . . .                                | 5        |
| 5.6      | Tabela: Trips . . . . .                                | 5        |
| 5.7      | Tabela: Trip_category . . . . .                        | 5        |
| <b>6</b> | <b>Uzasadnienie, że baza jest w EKNF</b>               | <b>6</b> |
| <b>7</b> | <b>Opis trudności podczas realizacji projektu</b>      | <b>6</b> |

## 1 Spis użytych technologii

- System zarządzania bazą danych: MariaDB
- Język programowania: Python (do generowania danych, spis bibliotek znajduje się w pliku `requirements.txt`)
- Narzędzie do analizy, generowania wykresów i raportowania: R Markdown (wszystkie biblioteki znajdują się w pliku `raport_team03`)
- Format raportu: PDF

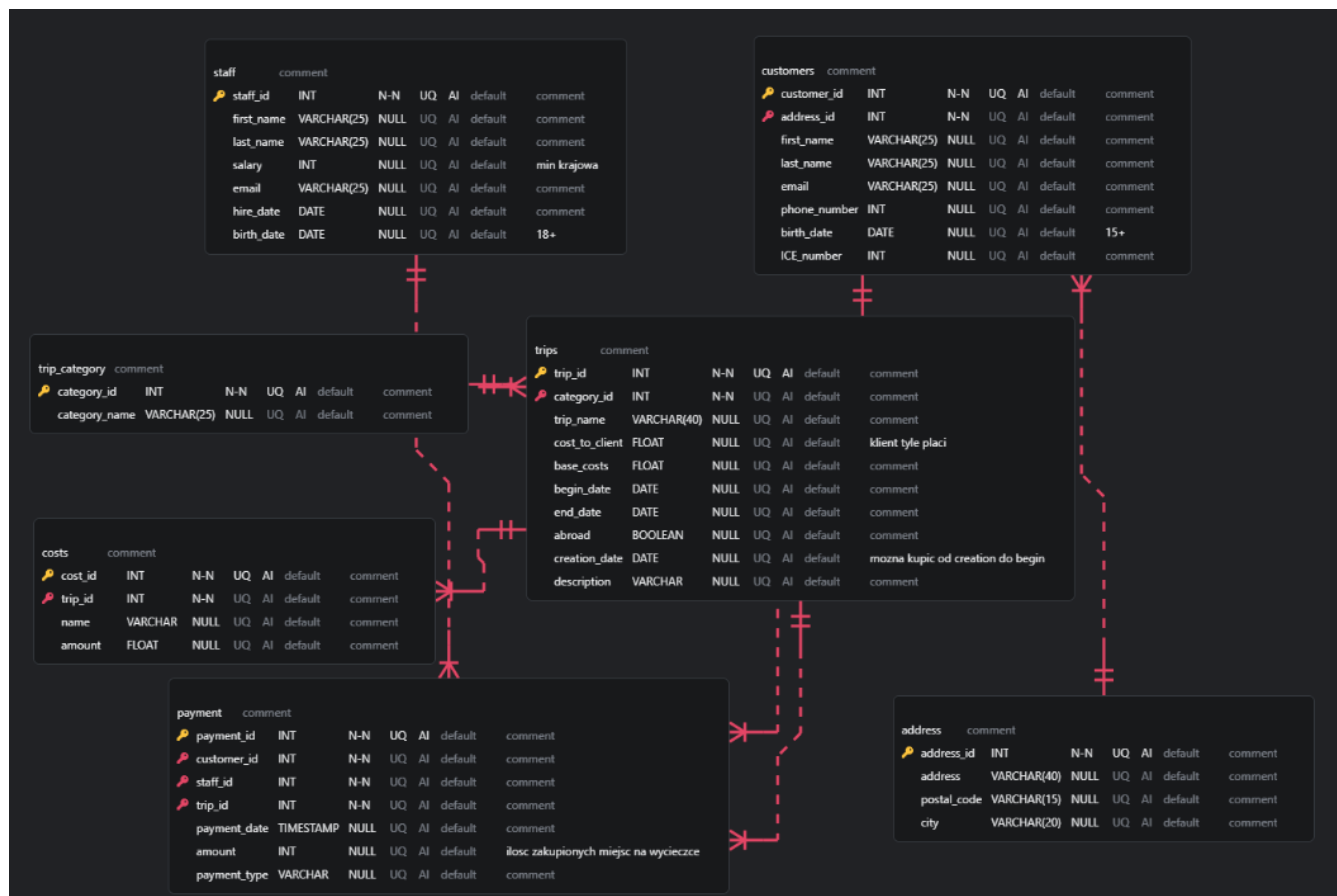
## 2 Lista plików i opis ich zawartości

- `category_fill.py` - Skrypt łączący się z bazą danych MySQL i wstawiający listę kategorii podróży,
- `generowanie_tabel.py` - Skrypt generujący dane i tabele potrzebne do bazy danych,
- `database_filler_template.py` - Skrypt wstawiający wygenerowane dane do bazy danych,
- `tabele.sql` - Plik sql tworzący potrzebne tabele,
- `tabele.py` - Skrypt wykonujący zapytania z `tabele.sql` na bazie danych,
- `wszystko.py` - Skrypt uruchamiający skrypty `tabele.py`, `category_fill.py` i `database_filler_template.py`,
- `requirements.txt` - Lista bibliotek Pythona i ich wersji, wymaganych do uruchomienia projektu,
- `raport_team03.Rmd` - Plik generujący raport,
- `raport_team03.pdf` - Raport w formacie PDF,
- `Adresy_StanNa20250119.csv` - Zbiór wszystkich adresów we Wrocławiu,
- `lista_imion.csv` - Zbiór polskich imion z częstością występowania,
- `lista_nazwisk.csv` - Zbiór polskich nazwisk z częstością występowania,
- `dokumentacja.pdf` - Dokumentacja projektu.

## 3 Kolejność i sposób uruchamiania plików

1. Otwieramy cały folder z plikami w VS Code
2. Pobieramy biblioteki komendą `pip install -r requirements.txt` poprzez terminal w VS Code
3. Uruchamiamy plik `wszystko.py`, aby utworzyć i wpisać nowe dane do tabeli
4. Otwieramy plik `raport_team03.Rmd` w RStudio i klikamy w opcję 'Knit'
5. Włączamy gotowy raport `raport_team03.pdf`

## 4 Schemat projektu bazy danych



Rysunek 1: Schemat bazy danych firmy Wombat Grylls sp. z o.o.

## 5 Lista zależności funkcyjnych dla każdej relacji

### 5.1 Tabela: Address

- address\_id → address, postal\_code, city

### 5.2 Tabela: Costs

- cost\_id → trip\_id, name, amount
- trip\_id → trip\_id (odwołanie do tabeli trips)

### 5.3 Tabela: Customers

- customer\_id → address\_id, first\_name, last\_name, email, phone\_number, amount, birth\_date, ICE\_number
- address\_id → address\_id (odwołanie do tabeli address)

### 5.4 Tabela: Payment

- payment\_id → customer\_id, staff\_id, trip\_id, payment\_date, amount
- customer\_id → customer\_id (odwołanie do tabeli customers)
- staff\_id → staff\_id (odwołanie do tabeli staff)

- trip\_id → trip\_id (odwołanie do tabeli trips)

### 5.5 Tabela: Staff

- staff\_id → address\_id, first\_name, last\_name, salary, email, hire\_date, birth\_date
- address\_id → address\_id (odwołanie do tabeli address)

### 5.6 Tabela: Trips

- trip\_id → category\_id, trip\_name, cost\_to\_client, begin\_date, end\_date, abroad, creation\_date, description
- category\_id → category\_id (odwołanie do tabeli trip\_category)

### 5.7 Tabela: Trip\_category

- category\_id → category\_name

## 6 Uzasadnienie, że baza jest w EKNF

Baza danych została zaprojektowana zgodnie z zasadami EKNF (Efektywnej Normalnej Formy Encji). W każdej tabeli spełnione są następujące warunki:

- Każda tabela posiada jednoznaczny klucz główny.
- Brak zależności przechodnich - wszystkie zależności funkcyjne są określone przez klucze obce.
- Zależności funkcyjne są zgodne z zasadami normalizacji, eliminując redundancję.

## 7 Opis trudności podczas realizacji projektu

Najtrudniejszym elementem było generowanie danych. Musieliśmy przyjąć pewne uproszczenia, które spowodowały, że baza nie jest tak realistyczna, jak mogłaby być w prawdziwym życiu.