

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

Autorzy

January 20, 2025

Contents

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

1 Spis użytych technologii

- System zarządzania bazą danych: MariaDB
- Język programowania: Python (do generowania danych)
- Narzędzie do analizy i raportowania: R Markdown
- Narzędzia do generowania wykresów: R Markdown
- Format raportu: PDF

2 Lista plików i opis ich zawartości

- `schemat.erd` - Diagram ERD bazy danych,
- `adresy.py` - Skrypt generujący losowe adresy przy pomocy biblioteki `geopy`,
- `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.

3 Kolejność i sposób uruchamiania plików

1.

4 Schemat projektu bazy danych

Figure 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