Dokumentacja

Amelia Dorożko, 282259
Matylda Mordal, 282240
Zuzanna Pawlik, 282230
Paweł Solecki, 282246
Zofia Stępień, 282254

29 stycznia 2025

1 Spis technologii

1.1 Użyte biblioteki i pakiety Python

- pylatex
- mysql.connector
- numpy
- matplotlib
- pandas
- faker
- random
- decimal
- datetime
- subprocess
- re
- unicodedata

1.2 Narzędzia i technologie zewnętrzne

- MySQL
- Python
- Jupyter Notebook
- Visual Studio Code
- LaTeX
- Conda

2 Lista plików

- **Schemat.json.vuerd** – Plik zawierający gotowy schemat bazy danych.
- **Raport.py** – Zawiera analizę danych i wykresy.
- **Dokumentacja.pdf** – Zawiera opis działania projektu oraz użytych technologii.
- **Uzupelnienie_tabel.ipynb** – Plik odpowiedzialny za wstawienie danych testowych do bazy.

3 Kolejność i sposób uruchamiania plików

3.1 Połączenie z bazą danych

- Na początku należy połączyć się z bazą.
- W oknie konfiguracji połączenia wprowadź odpowiednie dane:
  - **Login**: team19
  - **Hasło**: te@mzaiq
  - **Baza**: team19
- Po wpisaniu wszystkich danych kliknij przycisk "Connect". SQLTools nawiąże połączenie z bazą danych. Jeśli wszystkie dane zostały podane poprawnie, połączenie zostanie zrealizowane.

3.2 Schemat

Sposób uruchomienia:

- Na początku należy zainstalować rozszerzenie ERD Editor.
- Następnie pobierz plik schematu bazy danych w formacie *.json.vuerd*.
- Za pomocą ERD Editor przekształć na schemat SQL.
- Ostatecznie, uruchom wygenerowany kod w pliku *.sql*, będąc połączonym z bazą danych.

3.3 Uzupełnienie tabel

Sposób uruchomienia:

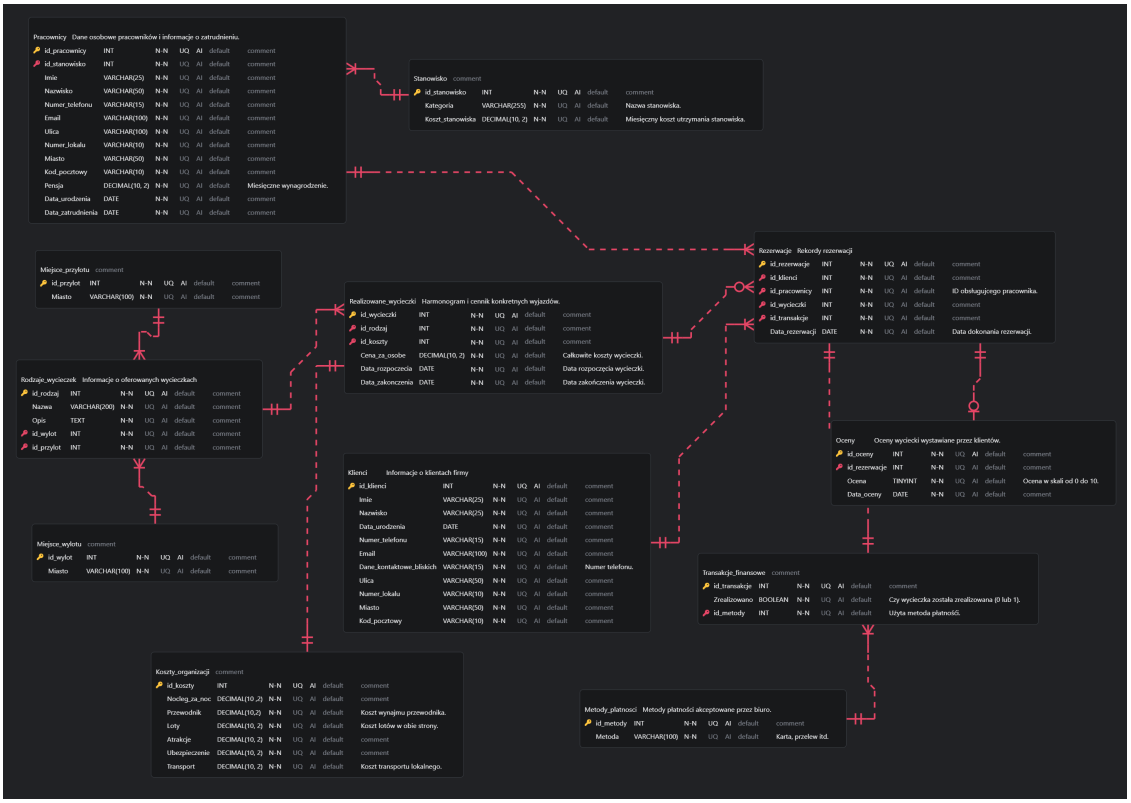
- Uruchom plik w środowisku Jupyter Notebook lub jako skrypt Python.
- Skrypt korzysta z biblioteki Faker do generowania przykładowych danych.
- Przed uruchomieniem upewnij się, że masz zainstalowane wymagane biblioteki: `pip install faker mysql-connector-python`.

3.4 Raport

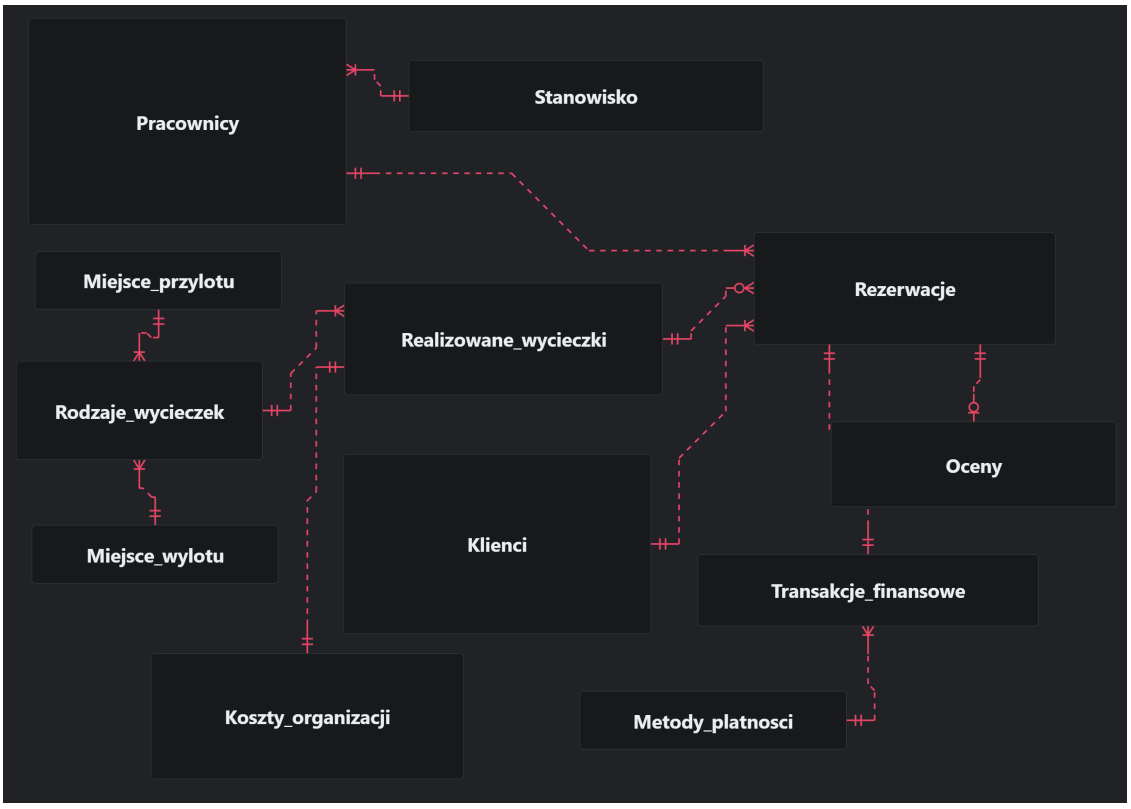
Sposób uruchomienia:

- Otwórz plik w Jupyter Notebook lub uruchom jako skrypt Python.
- Wykorzystuje biblioteki pandas, matplotlib, numpy, pylatex – upewnij się, że są zainstalowane: `pip install pandas matplotlib numpy pylatex`.

4 Schemat projektu bazy danych



Rysunek 1: Schemat bazy danych - wersja 1



Rysunek 2: Schemat bazy danych - wersja 2

5 Lista zależności funkcyjnych

- Tabela Klienci Klucz główny: id_klienci

  Zależności funkcyjne: id_klienci → Imie, Nazwisko, Data_urodzenia, Numer_telefonu, Email, Dane_kontaktowe_bliskich, Ulica, Numer_lokalu, Miasto, Kod_pocztowy

  Wszystkie atrybuty są w pełni zależne od klucza głównego.

- Tabela Pracownicy Klucz główny: id_pracownicy

  Zależności funkcyjne: id_pracownicy → id_stanowisko, Imie, Nazwisko, Numer_telefonu, Email, Ulica, Numer_lokalu, Miasto, Kod_pocztowy, Pensja, Data_urodzenia, Data_zatrudnienia id_stanowisko → Kategoria, Koszt_stanowiska

  Tranzytywna zależność id_pracownicy → id_stanowisko → Kategoria jest wydzielona poprzez tabelę Stanowisko, więc struktura jest poprawna.

- Tabela Rodzaje_wycieczek Klucz główny: id_rodzaj

  Zależności funkcyjne: id_rodzaj → Nazwa, Opis, id_wylot, id_przylot id_wylot → Miasto id_przylot → Miasto

  Tranzytywne zależności id_rodzaj → id_wylot → Miasto oraz id_rodzaj → id_przylot → Miasto są wydzielone poprzez tabele Miejsce_wylotu i Miejsce_przylotu.

- Tabela Realizowane_wycieczki Klucz główny: id_wycieczki

  Zależności funkcyjne: id_wycieczki → id_rodzaj, id_koszty, Cena_za_osobe, Data_rozpoczecia, Data_zakonczenia id_koszty → Nocleg_za_noc, Przewodnik, Loty, Atrakcje, Ubezpieczenie, Transport

  Zależności są poprawnie wydzielone poprzez tabelę Koszty_organizacji.

- Tabela Rezerwacje Klucz główny: id_rezerwacje

  Zależności funkcyjne: id_rezerwacje → id_klienci, id_pracownicy, id_wycieczki, id_transakcje, Data_rezerwacji Klucze obce (id_klienci, id_pracownicy, id_wycieczki, id_transakcje) odnoszą się do kluczy głównych w innych tabelach.

  Brak zależności funkcyjnych poza kluczami głównymi i obcymi.

- Tabela Koszty_organizacji Klucz główny: id_koszty

  Zależności funkcyjne: id_koszty → Nocleg_za_noc, Przewodnik, Loty, Atrakcje, Ubezpieczenie, Transport.

  Wszystkie atrybuty są w pełni zależne od klucza głównego.

- Tabela Transakcje_finansowe Klucz główny: id_transakcje

  Zależności funkcyjne: id_transakcje → Zrealizowano, id_metody id_metody → Metoda

  Tranzytywna zależność id_transakcje → id_metody → Metoda jest obsłużona przez tabelę Metody_platnosci.

- Tabela Oceny Klucz główny: id_oceny

  Zależności funkcyjne: id_oceny → id_rezerwacje, Ocena, Data_oceny id_rezerwacje → id_klienci, ...

  Zależności są poprawnie wydzielone poprzez tabelę Rezerwacje.

6 EKNF

Uzasadnienie spełnienia postaci EKNF :

- Każda tabela posiada pojedynczy klucz główny (np. ‘id_klienci‘, ‘id_koszty‘), który jest elementarny i jednoznacznie identyfikuje rekordy.

  *Przykład: Klucz ‘id_klienci‘ w tabeli ‘Klienci‘ determinuje wszystkie pozostałe atrybuty.

- Atrybuty niebędące kluczami zależą wyłącznie od klucza głównego, a nie od innych atrybutów.

  *Przykład: W tabeli ‘Klienci‘ atrybut ‘Email‘ nie wpływa na ‘Numer_telefonu‘.

- Wszystkie niebanalne zależności funkcyjne są oparte na kluczach głównych.

  *Przykład: W tabeli ‘Rezerwacje‘ klucz ‘id_rezerwacje‘ determinuje ‘id_klienci‘, ‘id_pracownicy‘ itd.

- Relacje między tabelami realizowane są przez klucze obce, które nie wprowadzają zależności częściowych ani przechodnich.

  *Przykład: ‘id_rodzaj‘ w tabeli ‘Realizowane_wycieczki‘ odnosi się do klucza głównego w ‘Rodzaje_wycieczek‘.

7 Najtrudniejsze elementy

Największym wyzwaniem okazało się zapewnienie spójności i realizmu danych czasowych w strukturze bazy. Konieczne było precyzyjne dostosowanie wartości dat do kontekstu oraz zachowanie odpowiednich zależności czasowych pomiędzy tabelami. Każdy wpis musiał tworzyć wiarygodną chronologię zdarzeń. Drugim istotnym problemem było opracowanie mechanizmu losowania danych. Konieczne było tworzenie zestawów zróżnicowanych danych przy zachowaniu ich merytorycznej poprawności.