

Dokumentacja projektu bazy danych

Karolina Bakalarz, Amelia Bieda, Agnieszka Staszekiewicz, Aleksandra Szczur, Paweł Wojarnik

30 czerwca 2025

1 Użyte technologie

- Python

- `mysql.connector` – służył do nawiązywania połączenia z bazą danych MariaDB oraz wykonywania zapytań SQL
- `pandas` – umożliwił przekształcanie i analizę danych pobranych z bazy w formie DataFrame
- `seaborn` – wykorzystany do tworzenia estetycznych wizualizacji statystycznych na podstawie danych z bazy
- `matplotlib.pyplot` – użyty do rysowania podstawowych wykresów i modyfikacji wyglądu elementów graficznych
- `matplotlib.colors.to_rgb` – umożliwił konwersję nazw kolorów na wartości RGB w celu dostosowania palety barw wykresów.
- `matplotlib.patches.Patch` – wykorzystany do tworzenia niestandardowych elementów legendy i oznaczeń na wykresach.
- `colorsys` – użyty do manipulowania przestrzeniami barw, np. przy konwersji między HLS a RGB w wizualizacji danych.
- `warnings` – użyty do tłumienia ostrzeżeń Pythona pojawiających się przy zapytaniach lub operacjach na danych.
- `IPython.display` – zostały użyte do czytelnego prezentowania wyników i wniosków w sformatowanej formie.
- `tabulate` – zostało użyte do estetycznego wyświetlania danych tabelarycznych w formacie tekstowym .

- SQL (MariaDB) – definicja i tworzenie schematu bazy

2 Struktura repozytorium

Tabela 1: Tabela opisująca strukturę repozytorium.

Plik	Typ pliku	Zawartość pliku
inicjalizacja_bazy.ipynb	IPYNB	Skrypt SQL wykonany w Pythonie służący do tworzenia i inicjalizacji bazy danych.
analiza_danych.ipynb	IPYNB	Notebook zawierający kody, wizualizacje oraz opis wyników.
analiza_danych.pdf	PDF	Końcowy wygenerowany raport z analizą.
dokumentacja.pdf	PDF	Plik zawierający dokumentację projektu.

3 Instrukcja uruchomienia

1. Inicjalizacja struktury bazy danych

Uruchom w terminalu poniższą komendę:

```
mysql -h giniewicz.it -P 3306 -u team03 -pte@mlae < tworzenie_bazy.sql
```

2. Wypełnienie bazy danymi

```
jupyter inicjalizacja_bazy.ipynb
```

3. Wygenerowanie raportu

```
jupyter nbconvert --to pdf --TemplateExporter.exclude_input=True analiza_danych.ipynb
```

albo (gdy powyższe nie działa)

```
jupyter nbconvert --to pdf --no-input analiza_danych.ipynb
```

4 Schemat projektu bazy danych

Poniżej przedstawiamy gotowy diagram schematu bazy danych.



Rysunek 1: Schemat bazy danych.

5 Zależności funkcyjne

Zależności funkcyjne to reguły, które opisują powiązania między kolumnami w tabelach bazy danych. Mówią one, że jeśli znamy wartość jednej kolumny (lub kilku), to jesteśmy w stanie jednoznacznie określić wartość innej kolumny (lub grupy kolumn).

Zależność funkcyjną zapisujemy w postaci:

$$X \rightarrow Y$$

gdzie:

- X — to kolumna (lub zestaw kolumn), od której zależy inna kolumna,
- Y — to kolumna (lub kolumny), których wartość jest jednoznacznie wyznaczana przez X .

Klucze główne to szczególny przypadek zależności funkcyjnych — zawsze determinują wszystkie pozostałe kolumny w tabeli. Dzięki analizie zależności możemy też łatwiej dostrzec, gdzie występują klucze obce i jakie są powiązania między tabelami.

W poniższych punktach przedstawiono zależności funkcyjne zidentyfikowane w poszczególnych tabelach naszej bazy danych. Poza kluczami głównymi (które zostały oznaczone w **ten sposób**) możemy zauważyć, w jakich tabelach mamy tę samą kolumnę, np. metoda_id pojawia się w tabeli transakcje oraz metody_platnosc.

Tabela: adresy

- adres_id \rightarrow adres_id, miasto, ulica_numer, kod_pocztowy

Tabela: alergeny

- alergen_id \rightarrow alergen_id, nazwa

Tabela: jedzenie

- jedzenie_id \rightarrow jedzenie_id, danie, liczba_kalorii, rodzaj, czy_vegetarianskie, czy_weganskie, opis

Tabela: jedzenie_alergen

- jedzenie_id \rightarrow jedzenie_id, alergen_id
- jedzenie_id \rightarrow jedzenie_id (tabela jedzenie)
- alergen_id \rightarrow alergen_id (tabela alergeny)

Tabela: kierunek

- kierunek_id \rightarrow kierunek_id, nazwa

Tabela: klienci

- klient_id \rightarrow klient_id, plec, imie, nazwisko, e_mail, telefon, data_urodzenia, adres_id, czy_vegetarian, czy_weganin, alergeny, waga, wzrost
- adres_id \rightarrow adres_id (tabela adresy)

Tabela: klient_alergen

- klient_id \rightarrow klient_id, alergen_id
- klient_id \rightarrow klient_id (tabela klienci)
- alergen_id \rightarrow alergen_id (tabela alergeny)

Tabela: kontakt_bliski

- b_klienta_id → b_klienta_id, klient_id, imie, nazwisko, relacja, telefon, email

Tabela: koszty_organizacji

- koszt_id → koszt_id, wyprawa_id, rodzaj_kosztu_id, kwota_pln
- rodzaj_kosztu_id → rodzaj_kosztu_id (tabela rodzaje_kosztow)

Tabela: metody_platnosci

- metoda_id → metoda_id, nazwa

Tabela: pojazdy

- pojazd_id → pojazd_id, nazwa, typ, maksymalny_udzwig_w_tonach, liczba_miejsc, data_produkcji, data_ostatniej_kontroli, producent

Tabela: pracownicy

- pracownik_id → pracownik_id, plec, imie, nazwisko, data_urodzenia, adres_id, stanowisko_id, staz_w_firmie, wynagrodzenie_miesieczne, email, telefon
- adres_id → adres_id (tabela adresy)
- stanowisko_id → stanowisko_id (tabela stanowiska)

Tabela: rodzaje_kosztow

- rodzaj_kosztu_id → rodzaj_kosztu_id, nazwa, min_kwota, max_kwota

Tabela: rodzaje_wypraw

- rodzaj_wyprawy_id → rodzaj_wyprawy_id, nazwa, opis, cena_wyprawy_mln, czas_trwania_dni, dodatkowe_aktywnosci, cena_aktywnosci_zl, kierunek_id
- kierunek_id → kierunek_id (tabela kierunek)

Tabela: stanowiska

- stanowisko_id → stanowisko_id, nazwa

Tabela: transakcje

- transakcja_id → transakcja_id, klient_id, wyprawa_id, data_transakcji, kwota_za_wyprawe_mln, czy_dod_atrakcja, kwota_atrakcji, metoda_id
- metoda_id → metoda_id (tabela metody_platnosci)

Tabela: uczestnicy_wyprawy

- klient_id → klient_id, wyprawa_id
- klient_id → klient_id (tabela klienci)
- wyprawa_id → wyprawa_id (tabela wyprawy)

Tabela: wyprawy

- wyprawa_id → wyprawa_id, data_startu, data_powrotu, pojazd_id, rodzaj_wyprawy_id, status

6 Uzasadnienie formy EKNF

Nasza baza danych została zaprojektowana w zgodzie z zasadami rozszerzonej postaci kluczowej (EKNF) i została opracowana z myślą o modelowaniu danych w praktycznych, złożonych systemach relacyjnych.

W EKNF dopuszczalne są niektóre zależności funkcyjne, których prawa strona nie jest kluczem kandydującym, pod warunkiem że nie wprowadzają redundancji ani anomalii aktualizacji. W naszym przypadku:

- Każda zależność funkcyjna ma po lewej stronie klucz główny lub jego nadzbiór, co spełnia wymóg EKNF.
- Nie występują anomalie wstawiania, usuwania ani aktualizacji, ponieważ dane są jednoznacznie identyfikowalne i dobrze zorganizowane dzięki normalizacji.
- Klucze obce są jasno określone i służą do odwzorowania relacji między bytami (np. adres_id, klient_id, wyprawa_id).

Z tego względu można uznać, że nasz projekt spełnia założenia EKNF, zapewniając jednocześnie wysoki poziom spójności logicznej i praktyczną użyteczność w analizie i rozwoju systemu.

7 Najtrudniejszy element realizacji projektu

W trakcie pracy nad naszym projektem napotkaliśmy różne problemy, jednak następujące były dla nas najtrudniejsze:

1. **Wypełnienie bazy danych** – Trzeba zadbać, aby daty, ceny, i relacje między danymi były spójne logicznie i chronologicznie. Dane muszą wyglądać realistycznie (np. nie wysyłać klientów na Marsa za 20 zł). Łatwo popełnić błąd, który sprawi, że baza będzie niespójna.
2. **Projektowanie schematu bazy danych zgodnego z EKNF** – Projektowanie schematu bazy zgodnego z EKNF wymagało starannego uporządkowania wszystkich zależności i relacji w sposób znormalizowany, jednocześnie dbając o przejrzystość oraz czytelność struktury bazy. Kluczowe było odpowiednie rozdzielenie danych oraz świadomy wybór kluczy głównych i obcych, w tym kluczy złożonych, aby zapewnić spójność i efektywność działania systemu.

8 Podział pracy

Podział pracy nad naszym projektem był następujący:

- **Projekt i utworzenie schematu** – Agnieszka Staszekiewicz, Paweł Wojarnik
- **Skryptowe wypełnienie bazy danych** – Agnieszka Staszekiewicz, Paweł Wojarnik, Aleksandra Szczur
- **Analizowanie danych** – Karolina Bakalarz, Amelia Bieda, Aleksandra Szczur, Agnieszka Staszekiewicz
- **Utworzenie raportu** – Agnieszka Staszekiewicz, Amelia Bieda
- **Utworzenie dokumentacji** – Karolina Bakalarz, Amelia Bieda