

Projekt zaliczeniowy - dokumentacja

Kacper Haczekiewicz, Maria Kwintal, Laura Nowak, Paweł Nowak

Styczeń 2025

1 Wstęp

Celem niniejszej dokumentacji jest szczegółowe przedstawienie komponentów projektu zaliczeniowego z przedmiotu Bazy Danych. Projekt dotyczy firmy Wombat Grylls sp. z o.o. organizującej wycieczki do miast, w których żyli znani matematycy. Dzięki niżej opisanej bazie danych możliwe jest przechowywanie i zarządzanie informacjami o miastach, klientach, pracownikach czy transakcjach związanych z firmą.

Dokumentacja zawiera opis wykorzystanych technologii, strukturę plików, graficzny schemat bazy danych oraz szczegółowy opis tabel i relacji pomiędzy nimi. Uwzględniono również podsumowanie pracy nad projektem.

2 Spis użytych technologii

Do realizacji projektu użyto następujących technologii.

1. Python 3.12 - użyty do napisania skryptów generujących dane i wypełniających bazę. Projekt jest również kompatybilny z wersją 3.11.
2. Biblioteki Pythona:
 - Mysql.connector 9.1.0 - umożliwia komunikację między Pythonem a bazą danych MySQL.
 - NumPy 2.1.3 - narzędzie do obliczeń numerycznych.
 - Pandas 2.2.3 - biblioteka do manipulacji i analizy danych.
3. ERD Editor 2.0.4 - wykorzystany do stworzenia schematu bazy danych w formie diagramu ERD.
4. R 4.4.2 - użyty do przygotowania raportu oraz analizy danych.
5. Pakiety R:
 - Dplyr - pakiet do manipulacji danych.
 - Ggplot2 - narzędzie do tworzenia wizualizacji danych.

- KableExtra - używany do formatowania tabel.
 - Knitr - pakiet do dynamicznego generowania raportów.
 - Plotly - umożliwia tworzenie interaktywnych wykresów.
 - RMariaDB - użyty do komunikacji z bazą danych MySQL w środowisku R.
6. LaTeX - użyty do przygotowania dokumentacji projektu.

3 Spis plików

Poniżej znajduje się opis plików zawartych w projekcie.

1. dokumentacja – katalog zawiera:
 - dokumentacja.tex – kod źródłowy LaTeX, użyty do przygotowania dokumentacji projektu.
 - dokumentacja.pdf - dokumentacja projektu w wersji PDF.
 - diagram.PNG – graficzny schemat bazy danych.
2. generator_danych.ipynb - skrypt Jupyter Notebook odpowiedzialny za generowanie danych. Dane te są zapisywane w katalogu Tabele_csv.
3. Imiona i nazwiska - katalog zawierający pliki CSV z danymi o popularnych imionach i nazwiskach, wykorzystywanymi przy generowaniu danych.
4. raport.Rmd - zawierający kod źródłowy R, który generuje raport.
5. schemat_bazy.vuerd - plik zawiera schemat bazy danych w formacie ERD.
6. Tabele_csv - katalog zawierający wygenerowane wcześniej pliki csv z danymi, które wstawiane są następnie do bazy danych.
7. Wstawiator_danych.py - skrypt w Pythonie odpowiedzialny za wstawianie danych z katalogu Tabele_csv do bazy danych.
8. wycieczki.csv - plik zawierający informacje o rodzajach wycieczek czy kosztach biletów.

4 Kolejność i sposób uruchamiania plików

Do uruchomienia projektu należy posiadać niezbędny sprzęt (tj. komputer z systemem Windows z wersją przynajmniej 10 i dostępem do internetu). Należy również posiadać Python w wersji przynajmniej 3.11, R 4.4.2, LaTeX oraz dowolne środowiska programistyczne pozwalające na kompilację bądź interpretację powyższych plików.

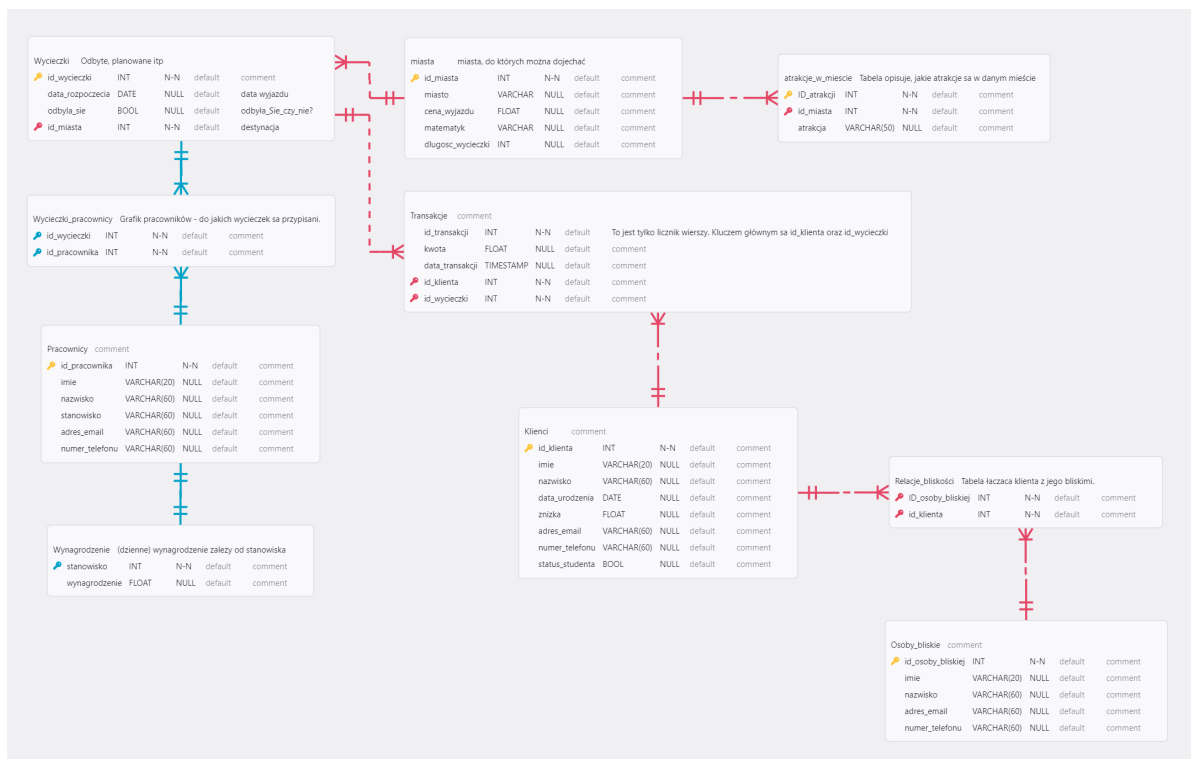
W celu uruchomienia projektu należy uruchomić następujące pliki w podanej kolejności.

1. generator_danych.ipynb
2. Wstawiator_danych.py
3. raport.Rmd

5 Schemat bazy danych

5.1 Graficzny schemat bazy

Poniżej znajduje się graficzny schemat bazy danych.



Rysunek 1: Schemat bazy

5.2 Opis tabel

Baza danych składa się z poniższych tabel.

1. Atrakcje_w_miescie
 - Atrybuty: id_atrakcji (klucz główny), atrakcja, id_miasta (klucz obcy).

- Zależności funkcyjne: $\text{id_atrakcji} \rightarrow \text{atrakcja, id_miasta}$.

2. Klienci

- Atrybuty: id_klienta (klucz główny), imie , nazwisko , data_urodzenia , znizka , adres_email , numer_telefonu , status_studenta .
- Zależności funkcyjne: $\text{id_klienta} \rightarrow \text{imie, nazwisko, data_urodzenia, znizka, adres_email, numer_telefonu, status_studenta}$.

3. Miasta

- Atrybuty: id_miasta (klucz główny), miasto , cena_wyjazdu , matematyk , dlugosc_wycieczki .
- Zależności funkcyjne: $\text{id_miasta} \rightarrow \text{miasto, cena_wyjazdu, matematyk, dlugosc_wycieczki}$.

4. Osoby_bliiskie

- Atrybuty: id_osoby_bliskiej (klucz główny), imie , nazwisko , adres_email , numer_telefonu .
- Zależności funkcyjne: $\text{id_osoby_bliskiej} \rightarrow \text{imie, nazwisko, adres_email, numer_telefonu}$.

5. Pracownicy

- Atrybuty: id_pracownika (klucz główny), imie , nazwisko , stanowisko (klucz obcy), adres_email , numer_telefonu .
- Zależności funkcyjne: $\text{id_pracownika} \rightarrow \text{imie, nazwisko, stanowisko}$ (klucz obcy), adres_email , numer_telefonu .

6. Relacje_bliiskosci

- Atrybuty: id_osoby_bliskiej (część klucza głównego), id_klienta (część klucza głównego).

7. Transakcje

- Atrybuty: id_transakcji (klucz główny), kwota , data_transakcji , id_klienta (klucz obcy), id_wycieczki (klucz obcy).
- Zależności funkcyjne: $\text{id_transakcji} \rightarrow \text{kwota, data_transakcji, id_klienta}$ (klucz obcy), id_wycieczki (klucz obcy).

8. Wycieczki

- Atrybuty: id_wycieczki (klucz główny), data_rozpoczecia , odbyla_sie , id_miasta (klucz obcy).
- Zależności funkcyjne: $\text{id_wycieczki} \rightarrow \text{data_rozpoczecia, odbyla_sie, id_miasta}$ (klucz obcy).

9. Wycieczki_pracownicy

- Atrybuty: id_wycieczki (część klucza głównego, klucz obcy), id_pracownika (część klucza głównego, klucz obcy).

10. Wynagrodzenie

- Atrybuty: stanowisko (klucz główny), wynagrodzenie.
- Zależności funkcyjne: stanowisko \rightarrow wynagrodzenie.

5.3 Opis relacji między tabelami

Pomiędzy tabelami występują następujące relacje.

1. Wycieczki i Wycieczki_pracownicy

- Relacja 1:n - jeden pracownik może być przypisany do wielu wycieczek.
- Klucz główny tabeli Wycieczki (id_wycieczki) występuje jako część klucza głównego w tabeli Wycieczki_pracownicy.

2. Wycieczki_pracownicy i Pracownicy

- Relacja 1:n - jedna wycieczka może mieć wielu pracowników.
- Klucz główny tabeli Pracownicy (id_pracownika) występuje jako część klucza głównego w tabeli Wycieczki_pracownicy.

3. Pracownicy i Wynagrodzenie

- Relacja 1:1 - jeden pracownik może mieć jedno wynagrodzenie.
- Klucz główny tabeli Wynagrodzenie (stanowisko) występuje jako klucz obcy w tabeli Pracownicy.

4. Wycieczki i Miasta

- Relacja 1:n - wiele wycieczek może odbywać się w jednym mieście.
- Klucz główny tabeli Miasta (id_miasta) jest kluczem obcym w tabeli Wycieczki.

5. Miasta i Atrakcje_w_miescie

- Relacja 1:n - w jednym mieście może być wiele atrakcji.
- Klucz główny tabeli Miasta (id_miasta) jest kluczem obcym w tabeli Atrakcje_w_miescie.

6. Wycieczki i Transakcje

- Relacja 1:n - jedna wycieczka może być związana z wieloma transakcjami.

- Klucz główny tabeli Wycieczki (`id_wycieczki`) występuje jako klucz obcy w tabeli Transakcje.

7. Transakcje i Klienci

- Relacja 1:n - jeden klient może być związany z wieloma transakcjami.
- Klucz główny tabeli Klienci (`id_klienta`) występuje jako klucz obcy w tabeli Transakcje.

8. Klienci i Relacje_bliskosci

- Relacja 1:n - jeden klient może mieć wiele przypisanych relacji z osobami bliskimi.
- Klucz główny tabeli Klienci (`id_klienta`) występuje jako klucz obcy w tabeli Relacje_bliskosci.

9. Relacje_bliskosci i Osoby_bliskie:

- Relacja 1:n - jedna osoba bliska może być powiązana z wieloma klientami.
- Klucz główny tabeli Osoby_bliskie (`id_osoby_bliskiej`) występuje jako klucz obcy w tabeli Relacje_bliskosci.

5.4 Uzasadnienie, że baza jest w EKNF

Baza danych spełnia założenia EKNF, ponieważ:

1. Dane przechowywane w tabelach są atomowe, a wiersze nie są powielone (czyli spełnia 1NF).
2. Wyeliminowano wszystkie częściowe zależności od jakiegokolwiek klucza potencjalnego (czyli spełnia 2NF).
3. Wszystkie atrybuty niekluczowe są zależne tylko i wyłącznie od klucza głównego, a nie od innych atrybutów niekluczowych (czyli spełnia 3NF).
4. Każda nietrywialna zależność funkcyjna ma postać, w której lewa strona jest nadkluczem lub prawa strona zawiera atrybut elementarny (czyli spełnia EKNF).

Zatem baza danych jest w EKNF.

6 Podsumowanie projektu

Podczas realizacji projektu wyzwaniem okazało się zaplanowanie i przygotowanie schematu bazy danych. Był to kluczowy etap pracy nad projektem, ponieważ na nim opierała się reszta jego realizacji. Wymagające było również generowanie

danych. Proces ten okazał się czasochłonny i wymagał zaawansowanej wiedzy z zakresu pisania skomplikowanych skryptów w Pythonie.

Pomimo różnych trudności projekt pozwolił nam znacząco rozwinąć umiejętności z zakresu pracy z bazami danych. Ponadto pozwolił nam rozwinąć się w pracy zespołowej, szczególnie w rozdzielaniu zadań oraz efektywnej współpracy. To doświadczenie może okazać się przydatne w przyszłej pracy zawodowej.