

1. Użyte technologie

- MySQL workbench
- Python
- Visual Studio Code

2. Lista plików i ich zawartość

W projekcie wykorzystywane są następujące pliki i skrypty:

Pliki danych (CSV):

1. **employee_salaries.csv** – zawiera informacje o wynagrodzeniach pracowników, w tym ich identyfikator (ID), datę zatrudnienia, datę oraz kwotę wypłaty.
2. **men_name.csv** i **women_name.csv** – listy imion męskich i żeńskich wraz z częstością ich występowania.
3. **men_surname.csv** i **women_surname.csv** – listy nazwisk męskich i żeńskich wraz z częstością ich występowania.
4. **wycieczki2.csv** – cennik wycieczek, uwzględniający różne warianty w zależności od liczby uczestników.
5. **zrealizowane_wycieczki.csv** – rejestr zrealizowanych wycieczek, zawierający datę zakupu, datę wyjazdu oraz identyfikator pracownika odpowiedzialnego za sprzedaż.

Pliki skryptów (Python):

1. **employee_salaries.py** – generuje wypłaty pracowników, obliczając wynagrodzenie podstawowe oraz prowizję na podstawie zrealizowanych wycieczek. Prowizje przypisywane są do miesięcy, w których odbyły się wyjazdy.
2. **generating_pesel.py** – łączy dane z plików **men_name.csv**, **women_name.csv**, **men_surname.csv**, **women_surname.csv**, tworząc listę losowych klientów (z imieniem, nazwiskiem, PESEL-em, numerem telefonu oraz adresem e-mail).
3. **generating_travels_data.py** – przypisuje każdemu wariantowi liczby uczestników odpowiednią wagę i losowo generuje listę wycieczek realizowanych w danym miesiącu.
4. **main3.py** – odpowiada za wypełnienie tabel oraz odczyt plików CSV.

Pliki związane z analizą danych:

1. **app_1.py** – generuje plik **data_1.json**, w którym są przechowywane dane do tabeli z informacją o najpopularniejszych wycieczkach.

2. **app_2.py** – generuje plik data_2.json, w którym przechowywane są dane, na których podstawie tworzony jest wykres ilości klientów w poszczególnych miesiącach roku 2024.
3. **app_3.py** – generuje plik data_3.json, w którym przechowywane są dane do tabeli informującej o wycieczkach, na które klienci najchętniej wracają.
4. **app_4.py** – generuje plik data_4.json, w którym są przechowywane dane do tabeli z listą miesięcy posortowanych według ilości rezerwacji malejąco.
5. **app_5.py** – generuje plik data_5.json, w którym są przechowywane dane o trzech najlepszych pracownikach roku 2024 (największa ilość wycieczek).
6. **app_6.py** – generuje plik data_6.json, w którym przechowywana jest informacja jaki jest średni czas oczekiwania na swoją wycieczkę.
7. **app_7.py** – generuje plik data_7.json, w którym przechowywane są dane na temat najsłabszych miesięcy dla firmy w roku 2024 (najmniejsze ilości rezerwacji).

3. Kolejność uruchamiania plików

1. Instrukcja.txt
2. main3.py
3. raport.pdf

4. Schemat projektu

Niniejszy projekt dotyczy działalności firmy zajmującej się organizacją ślubów za granicą. Firma oferuje szeroką gamę kierunków podróży, dostosowanych do potrzeb par młodych. Każda wycieczka posiada różne warianty, uwzględniające liczbę uczestników ceremonii.

W ramach realizacji projektu gromadzone są dane dotyczące:

- Zrealizowanych wycieczek,
- Kosztów operacyjnych firmy,
- Cennika usług,
- Informacji o klientach i pracownikach,
- Wynagrodzeń pracowników, uwzględniających zarówno podstawową pensję, jak i prowizję od sprzedaży wycieczek.

Dzięki analizie zgromadzonych danych możliwe jest określenie m.in.:

- Którzy pracownicy generują największe przychody dla firmy,
- Jakie wycieczki cieszą się największym zainteresowaniem,
- Sezonowości sprzedaży – w których miesiącach klienci najczęściej korzystają z usług.

5. Lista użytych funkcji

Projekt wykorzystuje szereg funkcji programistycznych do przetwarzania i analizy danych:

Funkcje obliczeniowe

- `calculate_commission` – oblicza prowizję na podstawie liczby uczestników wycieczki.
- `scale_salary` – skaluje wynagrodzenie powyżej 20 000 zł.

Funkcje przetwarzające dane osobowe klientów

- `read_csv` – oblicza liczbę osób o danym imieniu/nazwisku.
- `merge_two_csv` – łączy losowe imię z losowym nazwiskiem, tworząc nowego klienta.
- `calculate_weights` – przypisuje wagę imieniu na podstawie częstości jego występowania.
- `process_weights_data` – obsługuje błędy związane z przetwarzaniem wag.
- `divideListBasedOnGender` – tworzy dwie listy na podstawie płci.
- `calculate_surname_weights` – oblicza wagi nazwisk na podstawie ich częstości.
- `get_random_from_list` – losuje wartości z listy.
- `combine_names_and_surnames` – łączy imiona i nazwiska klientów.
- `generate_pesel_and_phone` – generuje numer PESEL oraz numer telefonu klienta i osoby kontaktowej.
- `generate_people` – tworzy pełny profil klienta, łącznie z adresem e-mail.
- `generate_phone_pool` – zapewnia unikalność numerów telefonów.
- `get_names_list` – generuje listę par młodych.

Funkcje związane z wycieczkami

- `losuj_liczbe_z_przedzialu` – losuje liczbę uczestników ślubu, uwzględniając wagę przypisaną częstości wyjazdów dla danej liczby osób.
- `generuj_wyjazdy_i_zakupy` – generuje daty wyjazdów i zakupów wycieczek.
- `zapisz_do_csv` – zapisuje przetworzone dane do plików CSV.

Funkcje związane z bazą danych

- `create_connection` – nawiązuje połączenie z bazą danych.
- `read_csv2` – wczytuje pliki CSV i przetwarza dane.

6. Baza w EKNF - uzasadnienie

Baza danych została zaprojektowana zgodnie z zasadami **EKNF** (Eliminacyjnej Kluczowej Normalnej Formy), co oznacza, że:

1. **Nie występują redundancje danych** – każda tabela zawiera unikalne informacje.
2. **Każda zależność funkcyjna zaczyna się od nadklucza lub kończy się na atrybucie elementarnym**, co eliminuje możliwość nadmiarowości danych.
3. **Każda tabela posiada unikalny identyfikator (ID)**, co pozwala na jednoznaczne powiązanie rekordów i uniknięcie powielania informacji.
4. **Brak zależności funkcyjnych pomiędzy tabelami** – tabele przechowujące dane dotyczące pracowników, klientów i wycieczek są od siebie niezależne.

7. Trudności przy realizacji projektu

Podczas implementacji projektu wystąpiły następujące trudności:

1. **Problemy z połączeniem z bazą danych na różnych komputerach** – różnice w konfiguracji środowiska oraz uprawnienia użytkowników wymagały dodatkowej konfiguracji.
2. **Poprawność i kompletność danych w tabelach** – konieczne było zapewnienie spójności oraz eliminacja ewentualnych błędów w danych wejściowych, takich jak brakujące wartości czy błędne formatowanie.
3. **Skalowanie wynagrodzeń i obliczanie prowizji** – wymagało to optymalizacji algorytmów, aby poprawnie uwzględniały okresy realizacji wycieczek oraz liczbę uczestników.