Dokumentacja
Projekt Bazy Danych

Natalia Lach 262303, Alicja Myśliwiec 262275,
Michał Bresiński 262302, Filip Oszczepaliński 262292

Matematyka Stosowana
Wydział Matematyki Politechniki Wrocławskiej

2023

Spis treści

1. Spis użytych technologii

Do stowrzenia projektu przydały się nastepujące technologie wraz z bibliotekami.

— Język Python (biblioteki wraz z wersjami zawarte są w pliku *requirements.txt*)
  1. SQLAlchemy
  2. PyMySQL
  3. pandas
  4. numpy
  5. matplotlib
  6. pygame

— Język R
  1. reticulate - biblioteka pozwala nam używać Python'a w programie RStudio

2. Lista plików wraz z zawartością

Struktura repozytorium:

— folder "Raport" zawierający pliki dotyczące raportu
  — podfolder "raport.Rmd" zawiera kod źródłowy raportu w formacie R Markdown
  — podfolder "raport.pdf" zawiera wygenerowany plik PDF raportu
— folder "data" zawierający dane
  — podfolder "csv" zawierający pliki CSV
    pliki CSV: "GRY.csv", "IK.csv", "IM.csv", "NK.csv", "NM.csv", "ULIC.csv"
  — podfolder "sound" zawierający pliki dźwiękowe
    plik dźwiękowy "outro-song_oqu8zAg.mp3"
  — plik "main.py" - plik źródłowy w języku Python
— folder "table" zawierający pliki związane z tabelami
  — plik "Create_table.sql" - plik SQL zawierający skrypt tworzenia tabel
  — plik "Tabele.vuerd.json" - plik JSON z definicją tabel w formacie Vuerd
— plik "README.md" - plik zawierający informacje o repozytorium
— plik "requirements.txt" - plik zawierający zależności wymagane do uruchomienia projektu

3. ***Jak uruchomić projekt?***

3.1. Przygotowanie środowiska

Przed próbą uruchomienia wszelkich skryptów, należy upewnić się, czy wszystkie języki programowania oraz wymagane biblioteki są zainstalowane. Używanymi językami programowania są *Python* oraz *R*. Biblioteki języka *Python* są wypisane w pliku "requirements.txt". Można je zainstalować za pomocą komendy w konsoli cmd *pip install -r requirements.txt*. Jedyną używaną biblioteką w języku *R* jest moduł "reticulate", który należy zainstalować ręcznie w konsoli *R* przy pomocy funkcji *install.packages("reticulate")*.

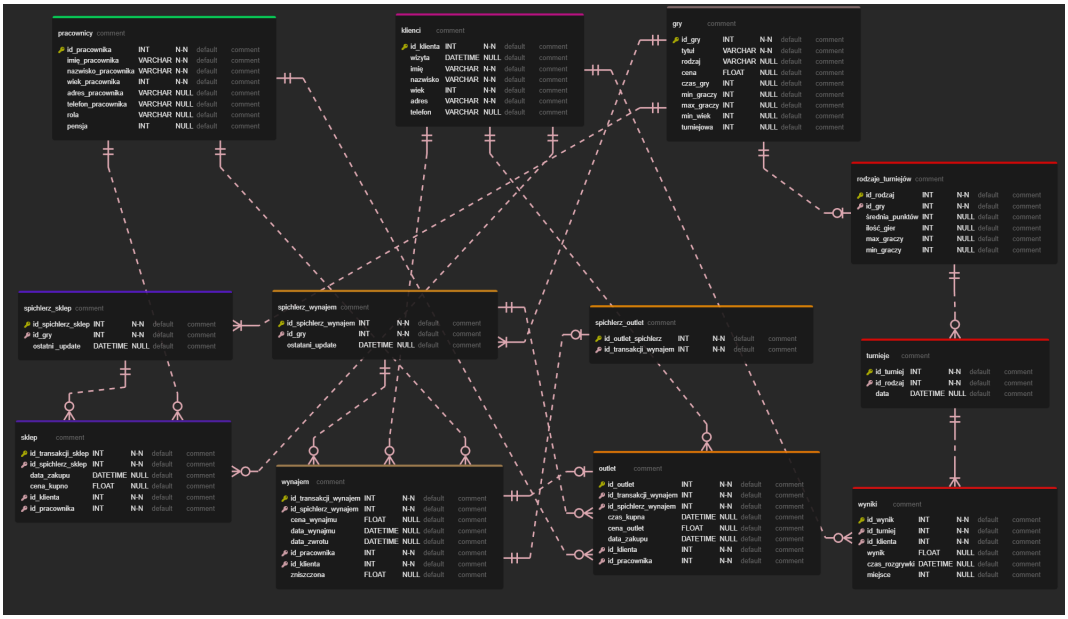
3.2. Generowanie tabel

Przechodząc do folderu **data** należy albo skorzystać z konsoli cmd i uruchomić skrypt za pomocą polecenia *python main.py*, bądź otworzyć plik za pomocą środowiska programistycznego typu Visual Studio Code i kliknąć *run code*. Tabele zostaną w tym momencie wysymulowane i dodane do bazy danych.
Po otrzymaniu powiadomienia "Database is ready", baza będzie gotowa do użycia (analizy).

3.3. Generowanie raportu

Po wygenerowaniu danych, należy powrócić do głównego folderu, gdzie znajduje się folder o nazwie **Raport**. Tam dostępna jest przykładowa wersja wygenerowanego raportu w formacie pdf, a także plik Rmd, który należy uruchomić, aby otrzymać aktualną wersję analizy na wygenerowanych wcześniej danych. Z racji rozszerzenia pliku, wymagane jest środowisko Rtudio oraz opcja Knit.

4. Schemat projektu bazy danych



Rys. 1: Schemat bazy danych

5. Lista zależności funkcyjnych dla relacji

1. relacja jeden do wielu między gry i spichlerz_sklep
   - w spichlerzu możemy mieć więcej egzemplarzy tej samej gry
   - dana gra ma przypisany jeden id w spichlerzu
2. relacja jeden do wielu między gry i spichlerz_wynajem
   - w spichlerzu możemy mieć więcej egzemplarzy tej samej gry
   - dana gra ma przypisany jeden id w spichlerzu
3. relacja jeden do wielu między spichlerz_sklep i sklep
   - zakupiona gra ma przypisany jeden id w spichlerzu
   - spichlerz jest odnawiany grami, którym przypisujemy stare spichlerzowe id
4. relacja jeden do wielu między klienci i sklep
   - klient może kupić wiele gier
   - dana gra może zostać kupiona tylko przez jednego klienta
5. relacja jeden do wielu między klienci i wynajem
   - klient może wypożyczyć wiele gier
   - dana gra może zostać wypożyczona tylko przez jednego klienta
6. relacja jeden do wielu między klienci i outlet
   - klient może kupić wiele gier
   - dana gra może zostać kupiona tylko przez jednego klienta
7. relacja jeden do wielu między pracownicy i sklep
   - pracownik może obsłużyć wiele transakcji

   — do danej transakcji jest przypisany jeden pracownik
8. relacja jeden do wielu między pracownicy i wynajem
   — pracownik może obsłużyć wiele transakcji
   — do danej transakcji jest przypisany jeden pracownik
9. relacja jeden do wielu między pracownicy i outlet
   — pracownik może obsłużyć wiele transakcji
   — do danej transakcji jest przypisany jeden pracownik
10. relacja jeden do wielu między gry i spichlerz_wynajem
   — w spichlerzu możemy mieć więcej egzemplarzy tej samej gry
   — dana gra ma przypisany jeden id w spichlerzu
11. relacja jeden do jednego między wynajem i outlet
   — id_transakcji jest unikatowe
   — zniszczony egzemplarz odpowiada tylko jednemu id_ transakcji
12. relacja jeden do wielu między spichlerz_wynajem i outlet
   — egzemplarz gry może trafić do outletu tylko raz
   — zniszczona gra jest uzupełniana nową grą o tym samym id, która następnie może znów się zniszczyć
13. relacja jeden do jednego między wynajem i spichlerz_outlet
   — egzemplarz gry może trafić do outletu tylko raz
   — id_transakcji_wynajem jest powiązane tylko z jedną grą
14. relacja jeden do jednego między gry a rodzaje_turniejów
   — gra może być turniejowa lub nie
   — rodzaj turnieju jest powiązany tylko z jedną grą
15. relacja jeden do jednego między rodzaj_turniejów i turnieje
   — jeden rodzaj turnieju może być rozegrany wiele razy
   — dany turniej ma przypisany tylko jeden rodzaj
16. relacja jeden do wielu między turnieje i wyniki
   — jeden turniej generuje wiele wyników
   — dany wynik jest przypisany do jednego turnieju
17. relacja między klienci i wyniki
   — jeden klient może brać udział w wielu turniejach
   — dany wynik jest przypisany do jednego klienta

6. EKNF

Baza danych opisana powyżej jest EKNF, ponieważ spełnia:

1. pierwszą postać normalną (1NF)
   — każda komórka zawiera wartości atomowe
   — wartości w kolumnach są tego samego typu
   — kolumny w tabelach mają unikalne nazwy
   — kolejność wierszy w tabelach nie ma znaczenia
2. drugą postać normalną (2NF)
   — baza jest w pierwszej postaci normalnej
   — brak zależności częściowych
3. trzecią postać normalną (3NF)
   — baza jest w drugiej postaci normalnej
   — brak zależności tranzytywnych
4. postać normalną klucza elementarnego (EKNF)
   — baza jest w trzeciej postaci normalnej
   — brak zależności wielowartościowych

7. Najtrudniejsza część realizacji projektu

Najtrudniejsze było ustalić jedną wspólną wersję projektu, ponieważ każdy miał na coś swój pomysł. Równie trudne okazało się wgranie danych do bazy zachowując przy tym relacje.