

Dokumentacja

1. Spis użytych technologii

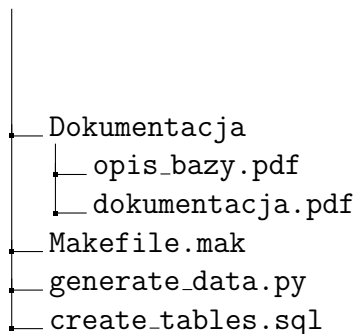
Do stworzenia projektu wykorzystane zostały poniższe technologie.

- Python z bibliotekami
 - pandas
 - numpy
 - random
 - urllib
 - sqlalchemy
 - datetime
 - unicode
- R z bibliotekami
 - RMariaDB
 - tidyverse
 - dbplyr
 - plotly
 - rmarkdown
- GNU Make
- Miro
- Overleaf

2. Lista plików i opis ich zawartości

W folderze `rental_games_shop` znajdują się pliki, które są ułożone według poniższego schematu.

```
rental_games_shop
├── data
│   ├── addresses.csv
│   ├── female_names.csv
│   ├── female_surnames.csv
│   ├── games_to_choose.csv
│   ├── male_names.csv
│   └── male_surnames.csv
└── Raport
    ├── raport.Rmd
    └── raport.html
```



W folderze `data` zawarte są pliki `.csv`, które zawierają dane użyte do generowania adresów, imion kobiet, nazwisk kobiet, imion mężczyzn, nazwisk mężczyzn i losowania gier, które będą dostępne sklepie i wypożyczalni.

W folderze `Raport` zawarte są dwa pliki związane z generowaniem raportu. W pliku `raport.Rmd` został wykorzystany pakiet R, który połączył się z wygenerowaną bazą danych i została tam wykonana analiza danych. Gotowy raport znajduje się w pliku `raport.html`.

W folderze `Dokumentacja` zostały zawarte dwa pliki `opis_bazy.pdf` oraz `dokumentacja.pdf`, które odpowiednio zawierają informacje o generowaniu bazy danych i dokumentację.

Plik `Makefile.mak` służy do generowania i wyświetlania raportu z wiersza poleceń `cmd`.

Plik `generate_data.py` służy do generowania bazy danych na podstawie podłączonych plików z folderu `Data` i zadanych parametrów.

Natomiast plik `create_tables.sql` służy do generowania pustych tabel, które zostaną zapełnione podczas automatycznego generowania bazy danych.

3. Kolejność i sposób uruchamiania plików, aby uzyskać gotowy projekt

Istnieje możliwość automatycznego generowania całej bazy danych. Do jej stworzenia potrzebny jest język `Python`, który musi być dodany do ścieżki zmiennych systemowych `PATH`. Aby wygenerować na nowo bazę danych należy przejść do odpowiedniej ścieżki i w konsoli `cmd` wywołać poniższą komendę.

```
C:\User\rental_games_shop>python generate_data.py
```

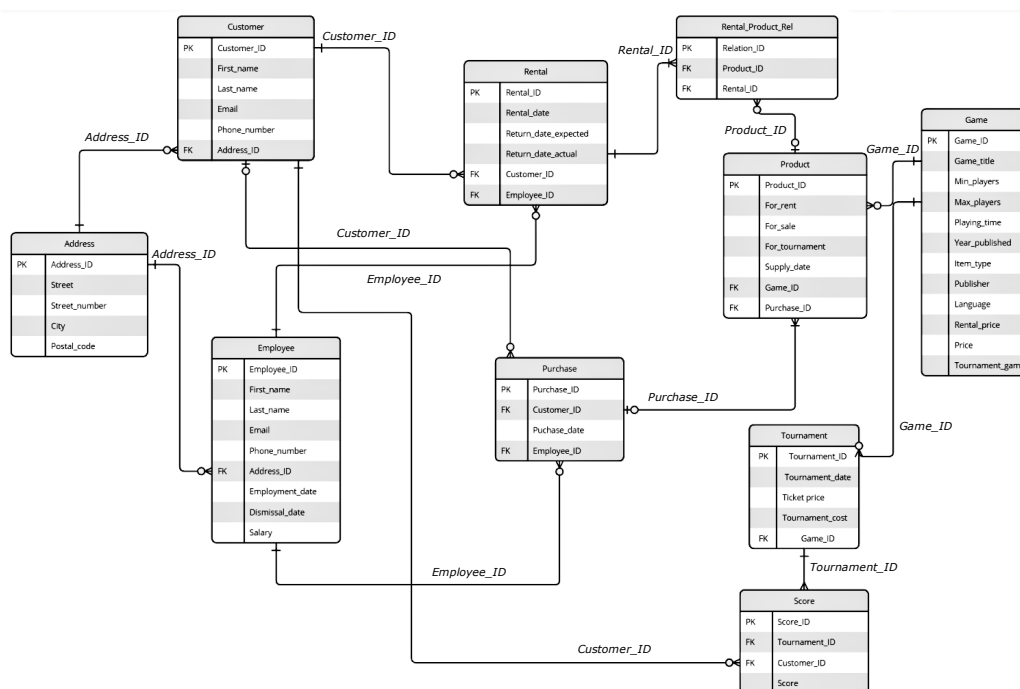
Powtórne generowanie całej bazy danych jest procesem czasochłonnym, więc do automatycznego generowania raportu będzie używana już wygenerowana baza. Aby automatycznie wygenerować raport należy w konsoli `cmd` wywołać poniższą komendę. W rezultacie automatycznie wygeneruje się raport w formacie `.html` i wyświetli.

```
C:\User\rental_games_shop>make open
```

Do stworzenia raportu potrzebny jest pakiet `R` i `Rscript`, które muszą być dodane do ścieżki ze zmiennymi systemowymi `PATH`. Raport tworzy się dzięki plikowi `Makefile.mak`.

4. Schemat projektu bazy danych i lista zależności funkcyjnych

Schemat bazy danych został zawarty na rysunku 1. Poszczególne łączenia między tabelami świadczą o istnieniu relacji między nimi.



Rysunek 1. Schemat bazy danych.

5. Uzasadnienie, że baza jest w EKNF

Stworzona baza danych jest EKNF, ponieważ spełnia wszystkie wymagane warunki.

- Pierwsza postać normalna (1NF)
 - każda tabela posiada własny klucz główny
 - nie występują powtórzenia grup informacji
 - kolejność wierszy jest dowolna, dane nie zależą od ich kolejności
 - każda wartość w bazie jest atomowa (niepodzielna)
- Druga postać normalna (2NF)
 - jest w pierwszej postaci normalnej
 - każda tabela zawiera dane dotyczące konkretnej klasy obiektów
- Trzecia postać normalna (3NF)
 - jest w drugiej postaci normalnej
 - atrybuty niekluczowe nie zależą przechodnio od żadnego z kluczy kandydujących
- Postać normalna klucza elementarnego (EKNF)
 - jest w trzeciej postaci normalnej
 - tabele nie zawierają zależności wielowartościowych

6. Opis, co było najtrudniejsze podczas realizacji projektu

Podczas realizacji projektu jednym z większych problemów było zaprojektowanie odpowiedniego schematu naszej bazy danych. Zależało nam, aby zaprojektowana baza była EKNF. Innym sporym wyzwaniem było automatyczne uzupełnienie bazy w taki sposób, aby dane wyglądały realistycznie.