

Projekt na bazy danych 2023 - Dokumentacja

Dominik O.

30 czerwca 2023

Spis treści

1	Spis użytych technologii	1
1.1	Python	1
1.2	MariaDb	1
1.3	Git	1
2	Lista plików i opis zawartości	2
3	Instrukcja uruchamiania	2
4	Schemat projektu bazy danych	3
5	Opis zależności funkcyjnych z wyjaśnieniem	3
5.1	Tabela <i>customers</i>	3
5.2	Tabela <i>games</i>	3
5.3	Tabela <i>inventory</i>	3
5.4	Tabela <i>list_of_players</i>	4
5.5	Tabela <i>staff</i>	4
5.6	Tabela <i>rental</i>	4
5.7	Tabela <i>payment</i>	4
5.8	Tabela <i>tournaments</i>	4
6	Trudności podczas tworzenia projektu	4

1 Spis użytych technologii

Podczas tworzenia projektu wykorzystane zostały różne narzędzia.

1.1 Python

Oraz biblioteki:

- **datetime**,
- **pandas** - wykorzystane podczas generowania dat,
- **mysql** - wykorzystana do łączenia się z bazą,

- **cvs** - wykorzystane do otwierania plików tekstowych,
- **math**,
- **numpy**,
- **random** - wykorzystane podczas generowania różnych danych,
- **itertools** - wykorzystane podczas analizy,
- **matplotlib** - wykorzystane podczas generowania wykresów,
- **os** - wykorzystane do nawigowania między plikami.

1.2 MariaDb

Jako dialekt MySQL użyty do komunikacji z bazą danych.

1.3 Git

Jako narzędzie do kontroli wersji i pracy w grupie.

2 Lista plików i opis zawartości

Projekt zawiera 4 skrypty w folderze głównym:

- *bob_the_builder.py*, który odpowiada za tworzenie, lub resetowanie bazy danych. Podczas tworzenia bazy uruchamia funkcje znajdujące się w pliku *fill.py*.
- *drop_all*, którym można usunąć wszystkie tabele z bazy danych.
- *Analiza_danych.py*, który analizuje bazę danych.
- *Generator.py*, który generuje raport na podstawie analizy.
- *dokumentacja.pdf*

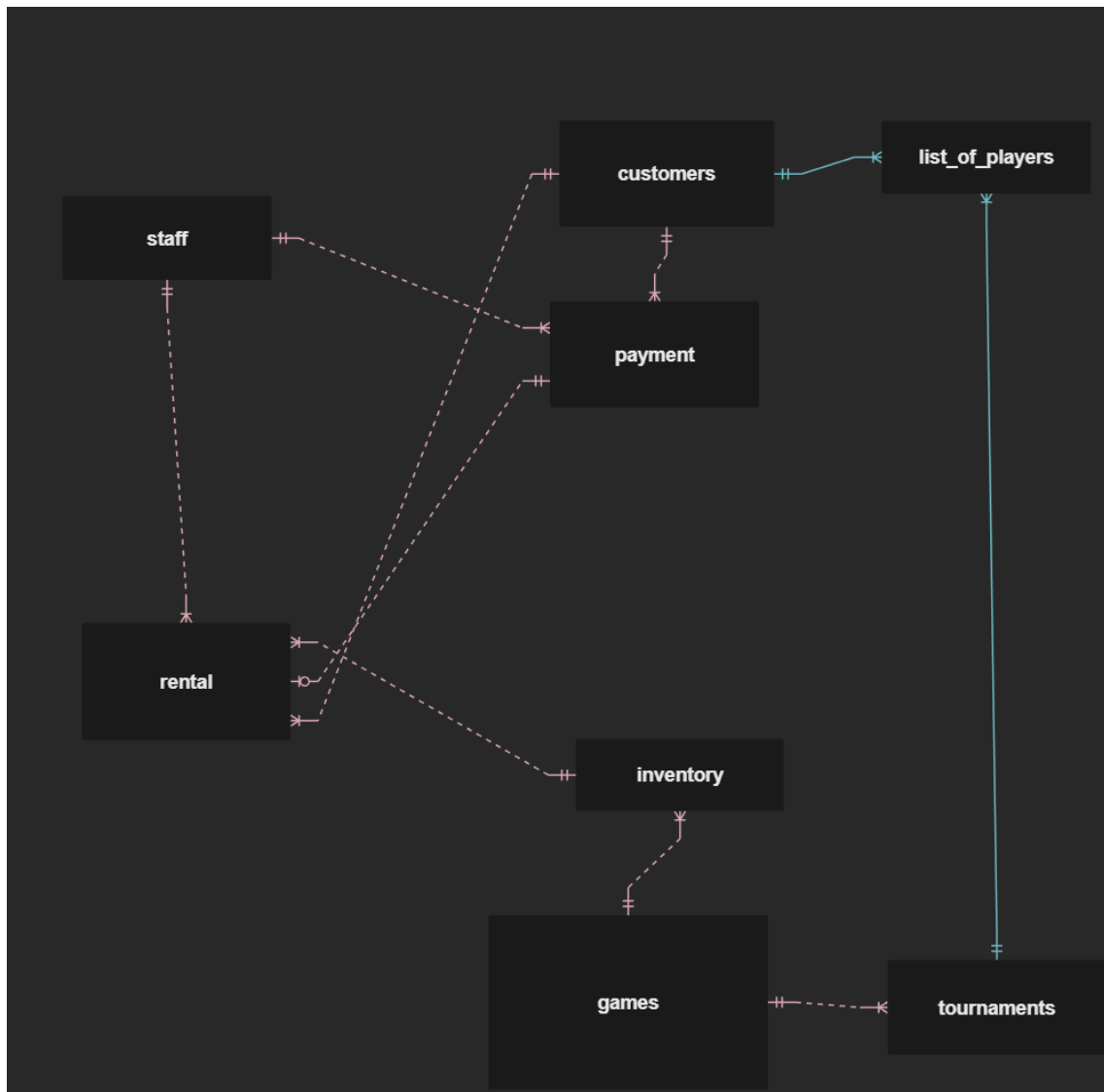
Oprócz tego w folderze głównym znajdują się foldery:

- *creators*, który zawiera moduły napisane w pythonie służące do tworzenia tabel w bazie.
- *generators*, który zawiera generatory losowych danych służące do uzupełniania danych w tabelach.
- *dokumentacja*, który zawiera pliki poboczne służące do generowania dokumentacji.

3 Instrukcja uruchamiania

By stworzyć nową bazę i wypełnić ją danymi należy uruchomić plik *bob_the_builder* (cały proces zajmuje niestety trochę czasu). By na podstawie danych znajdujących się obecnie w bazie wygenerować analizę należy uruchomić plik *Analiza_danych.py*. Plik generator tworzy pdf zawierający raport opraty na analizie.

4 Schemat projektu bazy danych



5 Opis zależności funkcyjnych z wyjaśnieniem

5.1 Tabela *customers*

Kluczem głównym tabeli jest *customer_id*.

5.2 Tabela *games*

Kluczem głównym jest *game_id*. Na podstawie id każdej z gier generowana jest tabela *inventory*. Korzystając z informacji o liczbie graczy i czasie gry z każdej z gier generowana jest tabela dotycząca turniejów.

5.3 Tabela *inventory*

Kluczem głównym jest *inventory_id*. *game_id* jest kluczem obcym odnoszącym się do *game_id* z tabeli *games*.

5.4 Tabela *list_of_players*

Tabela działa według założenia, że każdy gracz jest wpisywany do systemu jako klient by uprości zapis danych. Tabela posiada dwa klucze główne będące jednocześnie kluczami obcymi *customer_id* oraz *tournament_id* łącząc tym samym informacje o zawodnikach i turniejach.

5.5 Tabela *staff*

Kluczem głównym jest *staff_id*. Tabela nie posiada kluczy obcych.

5.6 Tabela *rental*

rental_id jest kluczem głównym tabeli. Tabela posiada 3 klucze obce *inventory_id*, *customer_id* oraz *staff_id* odnoszące się do kluczy głównych tabel *inventory*, *customer*, *staff*.

5.7 Tabela *payment*

payment_id jest kluczem głównym tabeli. Tabela posiada klucze obce *rental_id*, *customer_id* i *staff_id*.

5.8 Tabela *tournaments*

tournament_id jest kluczem głównym tabeli. Tabela posiada klcz obcy *game_id*. Dane w tabeli są generowane na podstawie gry, na której opraty jest dany turniej.

6 Trudności podczas tworzenia projektu

Największą trudnością było dogadanie się jak powinien wyglądać schemat bazy. Trudno było się też zorientować jak uzupełniać bazę za pomocą connectora do pythona. Dużo trudności sprawiło też sensowne stworzenie danych tak by tabele odnoszące się do siebie miały sens.