

BAZY DANYCH

# BAZA DANYCH GEEKS & DRAGONS

DOKUMENTACJA

Szymon Malec, 262276  
Adam Wrzesiński, 262317  
Michał Wiktorowski, 262330  
Weronika Zmyślona, 262284

Politechnika Wrocławska  
Wydział Matematyki - Matematyka Stosowana

# Contents

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Schemat bazy danych</b>	<b>2</b>
<b>3</b>	<b>Skryptowe wypełnienie bazy</b>	<b>3</b>
3.1	Tabela customers . . . . .	3
3.2	Tabela staff . . . . .	3
3.3	Tabela games_for_sale . . . . .	4
3.4	Tabela games_to_rent . . . . .	4
3.5	Tabela sale . . . . .	4
3.6	Tabela rental . . . . .	4
3.7	Tabela competition . . . . .	4
3.8	Tabela competition_results . . . . .	4
3.9	Połączenie z bazą . . . . .	4
<b>4</b>	<b>Analiza danych</b>	<b>4</b>
<b>5</b>	<b>Generowanie raportu</b>	<b>4</b>
<b>6</b>	<b>Technologie</b>	<b>4</b>
<b>7</b>	<b>Zarządzanie plikami</b>	<b>4</b>
<b>8</b>	<b>Baza w postaci EKNF</b>	<b>4</b>
<b>9</b>	<b>Podsumowanie</b>	<b>6</b>

# 1 Wstęp

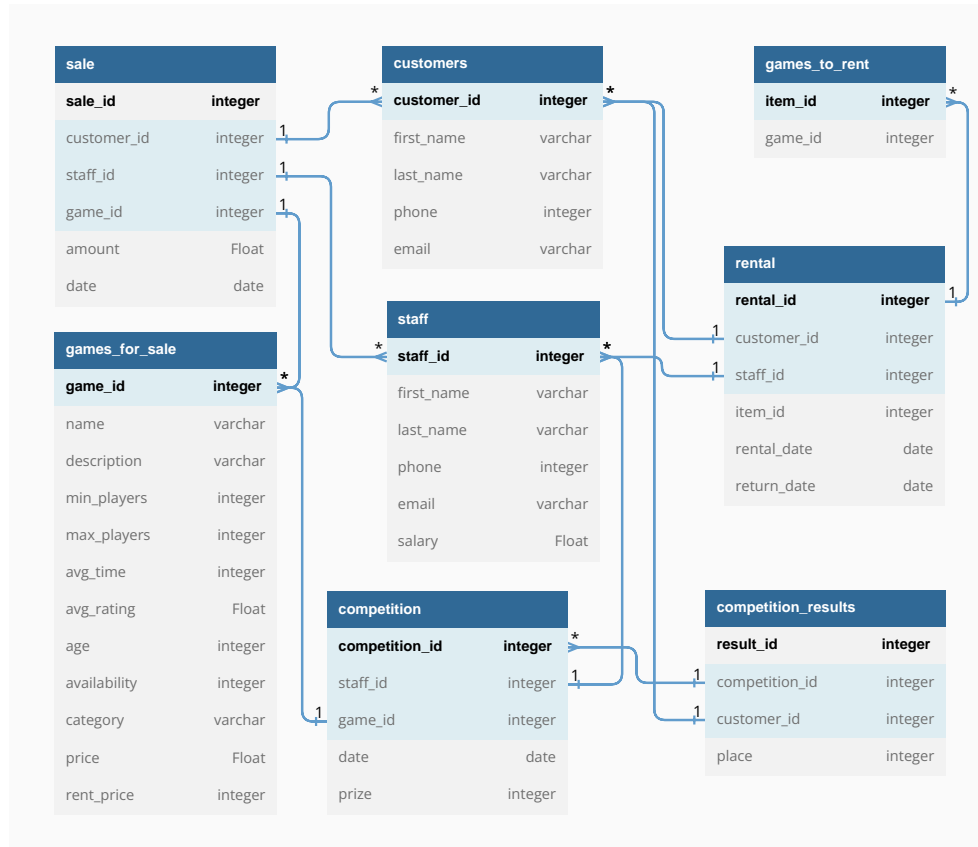
Niniejsza dokumentacja przedstawia proces tworzenia bazy danych sklepu Geeks & Dragons w ramach projektu na zaliczenie z kursu Bazy Danych z 2023 roku. Zadanie polegało na wysymulowaniu danych, jakie mogłyby się pojawić w bazie przykładowego sklepu, mającego w ofercie gry nieelektroniczne, który dodatkowo prowadzi wypożyczalnię gier oraz organizuje turnieje. Projekt został wykonany z wykorzystaniem języka programowania Python.

Pierwszą częścią projektu było wykonanie schematu bazy danych, a następnie wygenerowanie losowych danych. Uzyskane dane należało eksportować do bazy i umieścić na serwerze za pomocą MySQL. Kolejna część polegała na przeprowadzeniu analizy danych z wysymulowanego zbioru i przedstawienie jej w formie raportu. Ponadto został opisany przebieg normalizacji bazy danych do postaci EKNF.

## 2 Schemat bazy danych

Schemat bazy danych (Rysunek 1.) uwzględnia 8 tabel przechowujących następujące informacje:

- **customer** – dane dotyczące klientów sklepu,
- **staff** – dane dotyczące pracowników sklepu,
- **games\_for\_sale** – dane dotyczące gier dostępnych na sprzedaż,
- **games\_to\_rent** – dane dotyczące gier dostępnych do wypożyczenia,
- **sale** – dane na temat sprzedaży gier,
- **rental** – dane na temat wypożyczeń gier,
- **competition** – dane dotyczące organizowanych turniejów,
- **competition\_results** – dane dotyczące osiągniętych wyników przez uczestników turniejów.



Rysunek 1: Schemat bazy danych.

### 3 Skryptowe wypełnienie bazy

W celu przygotowania do wypełnienia bazy losowo wygenerowanymi danymi, zostały stworzone pliki csv odpowiadające poszczególnym tabelom z bazy danych. Oprócz danych generowanych losowo, pojawiają się również dane rzeczywiste, co zostanie opisane w dalszej części tego rozdziału. Na sam koniec, po połączeniu z serwerem, dane zostały zaimportowane do bazy.

#### 3.1 Tabela customers

Tabela customers zawiera informacje na temat 1329 klientów, które zorganizowane są w 5 następujących kolumnach:

- **customer\_id** – unikalne id dla każdego klienta,
- **first\_name** – imię klienta,
- **last\_name** – nazwisko klienta,
- **phone** – numer telefonu klienta,
- **email** – adres email klienta.

Dane w tabeli zostały wygenerowane w sposób losowy. Wartości w tabeli **customer\_id** wygenerowane zostały na pomocą odpowiedniego przedziału dodatnich liczb całkowitych. Imiona i nazwiska klientów zostały wylosowane z odpowiednim prawdopodobieństwem korzystając z danych zamieszczonych na stronie Głównego Urzędu Statystycznego. W pierwszej kolejności został wygenerowany wektor określający płeć klienta, z odpowiednim prawdopodobieństwem występowania danej płci, na podstawie liczby ludności we Wrocławiu z podziałem na mężczyzn i kobiety. Następnie w zależności od płci zostały przypisane imiona i nazwiska z odpowiednich tabeli danych pobranych ze wspomnianej strony internetowej.

Następna kolumna zawiera losowo generowane numery kontaktowe klientów. Numer w każdym przypadku składa się z połączenia losowo wybranego wyróżnika sieci telefonicznej (2 cyfry) oraz liczby z przedziału (1000000, 9000000). Lista dostępnych wyróżników została stworzona na podstawie informacji dostępnych na stronie Wikipedii. W ten sposób uzyskaliśmy indywidualny 9-cyfrowy numer dla każdego klienta.

Ostatnia kolumna zawiera adresy email, dla których identyfikatory użytkownika są generowane losowo na podstawie imion i nazwisk klientów. Identyfikatory adresów email składają się przykładowo:

- z połączenia imienia i nazwiska lub części nazwiska klienta,
- z połączenia imienia lub nazwiska i losowej liczby z przedziału (100, 10000).

Każdy adres email składa się również z domeny, która była losowana z listy domen stworzonej na podstawie rankingu najpopularniejszych serwisów pocztowych w Polsce dostępnego na stronie interaktywnie.com. Oczywiście, domeny były losowane z odpowiednim prawdopodobieństwem określonym według liczby użytkowników korzystających z wybranych serwisów pocztowych.

#### 3.2 Tabela staff

Tabela staff zawiera 6 kolumn z następującymi informacjami na temat 17 pracowników sklepu:

- **staff\_id** – zawiera unikalne id dla każdego pracownika,
- **first\_name** – imię pracownika,
- **last\_name** – nazwisko pracownika,
- **phone** – numer telefonu pracownika,
- **email** – adres email pracownika,
- **salary** – miesięczne wynagrodzenie pracownika (cena brutto).

Dane w pierwszych 4 kolumnach zostały wygenerowane w sposób analogiczny jak w przypadku tabeli customers. Nastąpiła zmiana w generowaniu adresów email dla pracowników. Są one tworzone w jednakowy sposób dla każdego pracownika – składają się z połączenia imienia i nazwiska za pomocą kropki oraz odrębnej domeny firmy @dragons.com.

Ostatnia kolumna określa miesięczne wynagrodzenie pracownika, które były losowane z listy konkretnych pensji [4310, 5140, 6530, 7280, 8320], które zostały wybrane na podstawie możliwych zarobków na stanowiskach dotyczących obsługi klienta. Losowanie odbyło się z ustalonym prawdopodobieństwem, zakładającym, że doświadczonych pracowników o większych zarobkach jest mniej.

### 3.3 Tabela games\_for\_sale

### 3.4 Tabela games\_to\_rent

### 3.5 Tabela sale

### 3.6 Tabela rental

### 3.7 Tabela competition

### 3.8 Tabela competition\_results

### 3.9 Połączenie z bazą

Tutaj opisać co dzieje się w pliku Connection.ipynb

## 4 Analiza danych

Opis jak została wykonana analiza danych, jakie pytania zostały postawione...

## 5 Generowanie raportu

W jaki sposób generowany jest raport...

## 6 Technologie

Korzystanie z bibliotek pandas, sqlalchemy itp...

## 7 Zarządzanie plikami

Lista plików i opis ich zawartości + kolejność i sposób urachamiania plików, aby uzyskać gotowy projekt

## 8 Baza w postaci EKNF

W tej części dokumentacji zawarto listę zależności funkcyjnych dla każdej relacji oraz uzasadnienie, że baza jest w EKNF.

### Tabela games\_for\_sale

Zależność funkcyjna: {game\_id, name, description} → pozostałe kolumny

Uzasadnieniem jest fakt, że game\_id jest kluczem głównym tabeli, a name i description są wartościami unikalnymi dla każdego game\_id, więc każda pozostała kolumna zależy funkcyjnie od zbioru tych trzech atrybutów. Jest to więc nietrywialna zależność funkcyjna, która zaczyna się od nadklucza, a więc nie zaburza to postaci EKNF bazy.

### **Tabela games\_to\_rent**

Zależność funkcyjna:  $\text{item\_id} \rightarrow \text{game\_id}$

Uzasadnieniem jest fakt, że  $\text{item\_id}$  jest kluczem głównym tabeli, więc  $\text{game\_id}$  zależy funkcyjnie od niego. Relacja ta nie zaburza postaci EKNF bazy, ponieważ tabela `games_to_rent` nie posiada żadnych niekluczowych zależności funkcyjnych.

### **Tabela sale**

Zależność funkcyjna:  $\text{sale\_id} \rightarrow \text{pozostałe kolumny}$

Uzasadnieniem jest fakt, że  $\text{sale\_id}$  jest kluczem głównym tabeli, więc każda pozostała kolumna zależy funkcyjnie od niego. Relacja ta nie zaburza postaci EKNF bazy, ponieważ tabela `games_to_rent` nie posiada żadnych niekluczowych zależności funkcyjnych.

### **Tabela rental**

Zależność funkcyjna:  $\text{rental\_id} \rightarrow \text{pozostałe kolumny}$

Uzasadnieniem jest fakt, że  $\text{sale\_id}$  jest kluczem głównym tabeli, więc każda pozostała kolumna zależy funkcyjnie od niego. Relacja ta nie zaburza postaci EKNF bazy, ponieważ tabela `games_to_rent` nie posiada żadnych niekluczowych zależności funkcyjnych.

### **Tabela customers**

Zależność funkcyjna:  $\{\text{customer\_id phone, email}\} \rightarrow \text{pozostałe kolumny}$

Uzasadnieniem jest fakt, że  $\text{customer\_id}$  jest kluczem głównym tabeli, a  $\text{phone}$  i  $\text{email}$  są wartościami unikalnymi dla każdego  $\text{customer\_id}$ , więc każda pozostała kolumna zależy funkcyjnie od zbioru tych trzech atrybutów. Jest to więc nietrywialna zależność funkcyjna, która zaczyna się od nadklucza, a więc nie zaburza to postaci EKNF bazy.

### **Tabela staff**

Zależność funkcyjna:  $\{\text{staff\_id phone, email}\} \rightarrow \text{pozostałe kolumny}$

Uzasadnieniem jest fakt, że  $\text{staff\_id}$  jest kluczem głównym tabeli, a  $\text{phone}$  i  $\text{email}$  są wartościami unikalnymi dla każdego  $\text{staff\_id}$ , więc każda pozostała kolumna zależy funkcyjnie od zbioru tych trzech atrybutów. Jest to więc nietrywialna zależność funkcyjna, która zaczyna się od nadklucza, a więc nie zaburza to postaci EKNF bazy.

### **Tabela competition**

Zależność funkcyjna:  $\text{competition\_id} \rightarrow \text{pozostałe kolumny}$

Uzasadnieniem jest fakt, że  $\text{competition\_id}$  jest kluczem głównym tabeli, więc każda pozostała kolumna zależy funkcyjnie od niego. Relacja ta nie zaburza postaci EKNF bazy, ponieważ tabela `games_to_rent` nie posiada żadnych niekluczowych zależności funkcyjnych.

### **Tabela competition\_results**

Zależność funkcyjna:  $\text{result\_id} \rightarrow \text{pozostałe kolumny}$

Uzasadnieniem jest fakt, że  $\text{result\_id}$  jest kluczem głównym tabeli, więc każda pozostała kolumna zależy funkcyjnie od niego. Relacja ta nie zaburza postaci EKNF bazy, ponieważ tabela `games_to_rent` nie posiada żadnych niekluczowych zależności funkcyjnych.

Podsumowując, baza jest w EKNF, ponieważ każda z występujących w niej relacji nie zaprzecza tej postaci.

## 9 Podsumowanie

M.in. co było najtrudniejsze podczas realizacji projektu