



Uniwersytet Gdański  
Wydział Matematyki, Fizyki i Informatyki  
Instytut Informatyki

# Projekt 7 - Baza danych gry RPG

Marcin Juozulinas

Projekt z przedmiotu bazy danych na kierunku informatyka profil ogólnoakademicki na Uniwersytecie Gdańskim.

Gdańsk  
25 maja 2020

## Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
<b>2</b>	<b>Opis projektu</b>	<b>2</b>
2.1	Potencjalne grupy użytkowników . . . . .	2
2.2	Wymagania funkcjonalne . . . . .	2
2.3	Wymagania нефункционалне . . . . .	2
2.4	Diagram związków encji . . . . .	3
<b>3</b>	<b>Przykłady realizacji bazy danych</b>	<b>5</b>
3.1	Przykłady zawartości najważniejszych tabel . . . . .	6
3.2	Przykłady kilku zapytań i ich wyników . . . . .	7

# 1 Wprowadzenie

Baza danych przeznaczona jest do gromaczenia danych na potrzeby prostej gry mmo. Głównym jej założeniem jest umożliwienie przechowywania danych na temat kont i postaci użytkowników.

## 2 Opis projektu

Projekt powstał na zlecenie firmy, która potrzebuje bazy danych na potrzeby gry MMO. Baza danych powstała wedle kryteriów narzuconych przez daną firmę.

### 2.1 Potencjalne grupy użytkowników

- Administrator – główny zarządca bazy danych, posiada pełen dostęp do bazy danych
- Projektant bazy danych - odpowiedzialny za logiczny i fizyczny projekt bazy danych
- Użytkownik - komunikuje się z bazą danych za pomocą zewnętrznych programów np. w celu sprawdzenia stanu konta lub historii zamówień

### 2.2 Wymagania funkcjonalne

Baza danych będzie przechowywać informacje o kontach założonych przez użytkowników, zrealizowanych zamówieniach i postaciach stworzonych w świecie gry.

System będzie aktualnie śledził i aktualizował statystyki gracza, jego położenie na mapie gry oraz posiadane przedmioty i umiejętności.

### 2.3 Wymagania niefunkcjonalne

Baza danych została zrealizowana przy użyciu MySQL Server 8.0.20 i MySQL Workbench 8.0.20 dla systemów windows.

Wady MySQL:

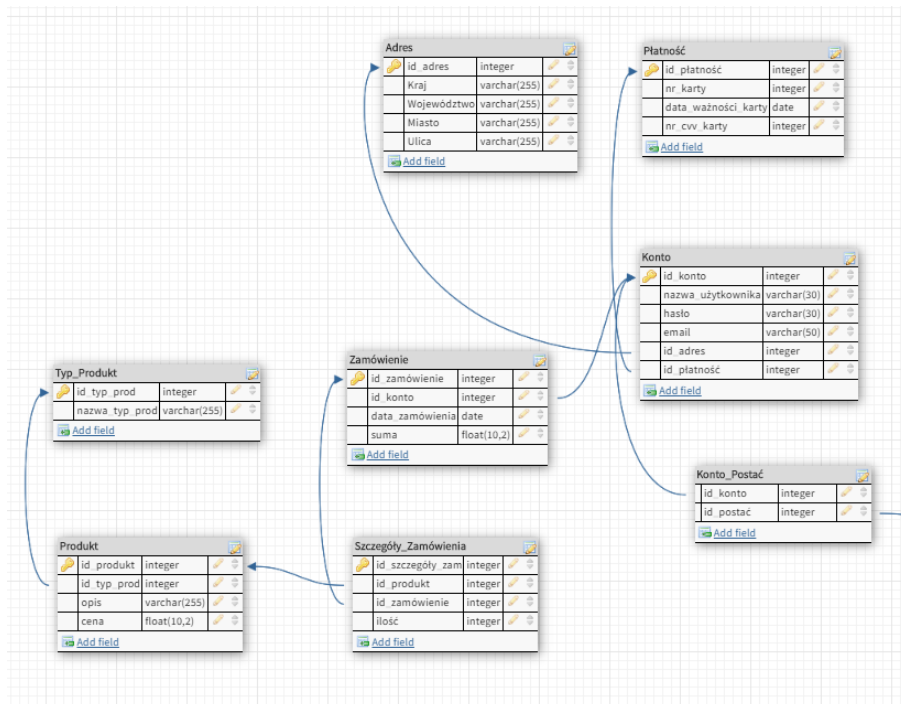
- MySQL nie obsługuje bardzo dużych rozmiarów baz danych wydajnie
- Istnieją problemy ze stabilnością
- Ma słabe skalowanie wydajności.

Zalety MySQL:

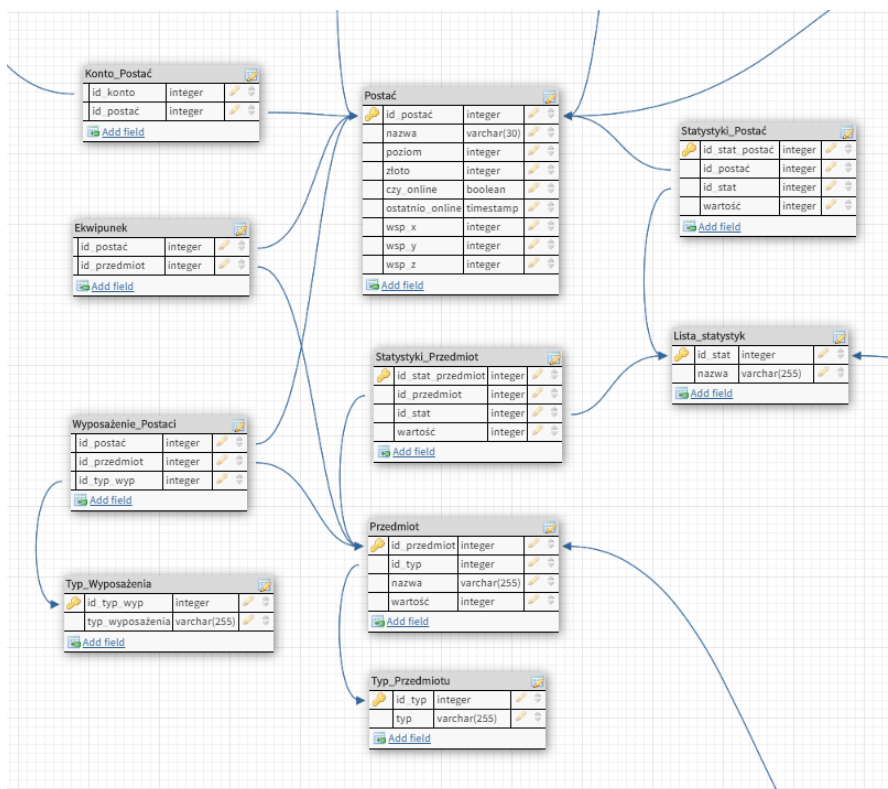
- Ochrona danych
- Jest łatwy w obsłudze

## 2.4 Diagram związków encji

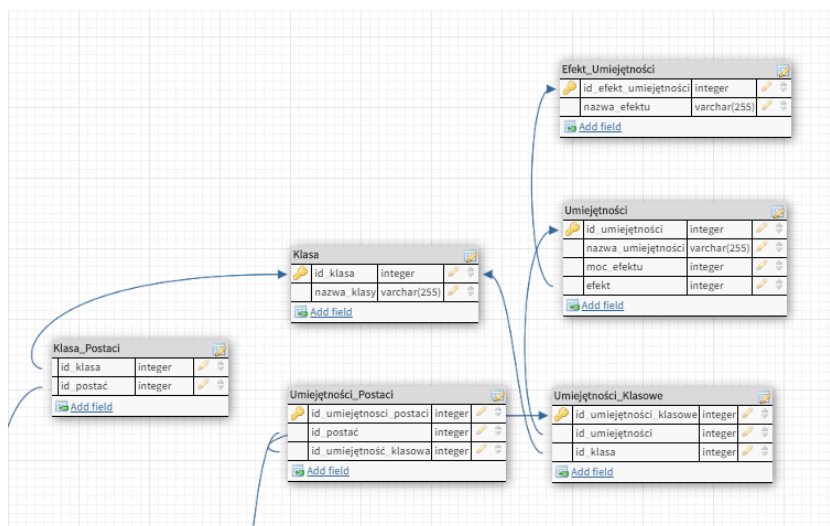
Poniżej znajduje się schemat ERD projektu podzielony na części z krótkimi opisami.



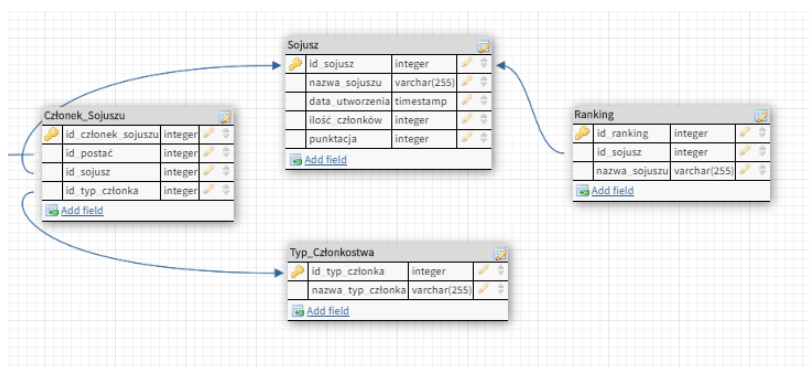
Fragment schematu ERD zawierający tabele z informacjami o kontach użytkowników i złożonymi przez nich zamówieniami.



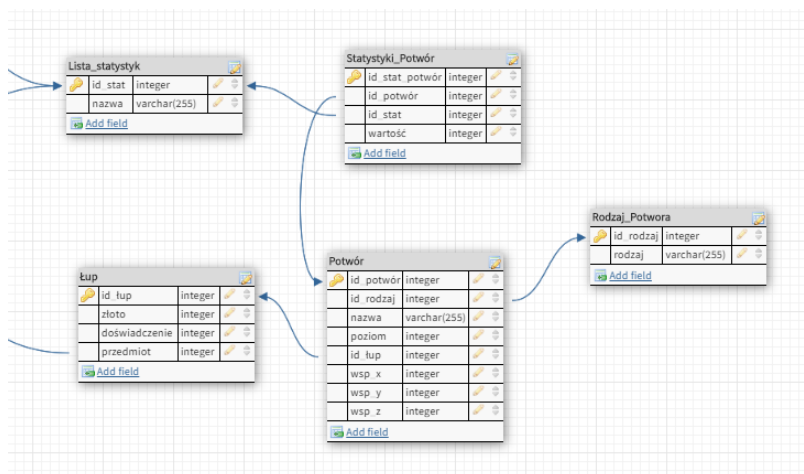
Główna część schematu ERD zawierająca tabele Postać, która łączy ze sobą wiele tabel i przechowuje informacje o postaci danego użytkownika. np. jego ekwipunek, statystyki i umiejętności.



Tabele zawierające informacje o klasach i umiejętnościach dostępnych w grze.



Tabele istniejących sojuszy i ich członków.



Tabele zawierające informacje o potworach, ich statystykach i możliwych łupach.

### 3 Przykłady realizacji bazy danych

Jedne z najważniejszych tabel to **Konto**, **Postać** i **Potwór**. Zawierają one kolejno dane użytkownika, informacje o jego postaci/postaciach i informacje o potworach występujących w grze.

#### **Konto:**

- id\_konto - klucz główny tabeli.
- nazwa\_użytkownika - pole typu varchar zawierające login użytkownika do 30 znaków.
- hasło - pole typu varchar zawierające hasło użytkownika.
- email - pole typu varchar zawiera adres email użytkownika.
- id\_adres - klucz obcy łączący tabele konto z tabelą adres zawierającą szczegółowe informacje o adresie użytkownika.
- id\_płatność - klucz obcy łączący tabele konto z tabelą płatność zawierającą dane o metodach płatności potrzebnych do realizacji zamówień.

#### **Postać:**

- id\_postać - klucz główny tabeli.
- nazwa - pole typu varchar zawierające nazwe postaci.
- poziom - pole typu int zawierające informacje o aktualnym poziomie postaci.
- złoto - pole typu int zawierające informacje o posiadanej ilości złota(waluty) na danej postaci.
- czy\_online - pole typu bool zwracające wartość true gdy postać jest online.
- ostatnio\_online - pole typu timestamp aktualizowane przy modyfikacji pola czy\_online. Kiedy użytkownik się wyloguje i pole czy\_online zmieni się z true na false.
- wsp\_x - pole typu int zawierające informacje o aktualnym położeniu postaci.
- wsp\_y - pole typu int zawierające informacje o aktualnym położeniu postaci.
- wsp\_z - pole typu int zawierające informacje o aktualnym położeniu postaci.

**Konto:**

- id\_potwór - klucz główny tabeli.
- id\_rodzaj - klucz obcy łączący tabele potwór z tabelą rodzaj\_potwora w której opisane są różne typy potworów.
- nazwa - pole typu varchar zawierające nazwe potwora.
- poziom - pole typu int zawierające informacje o poziomie danego potwora.
- id\_hup - klucz obcy łączący tabele potwór z tabelą hup zawierającą informacje o ilości doświadczenia, złota i możliwym przedmiocie otrzymanym po pokonaniu przeciwnika.
- wsp\_x - pole typu int zawierające informacje o aktualnym położeniu potwora.
- wsp\_y - pole typu int zawierające informacje o aktualnym położeniu potwora.
- wsp\_z - pole typu int zawierające informacje o aktualnym położeniu potwora.

### 3.1 Przykłady zawartości najważniejszych tabel

tabela **Konto**:

Field	Type	Null	Key	Default	Extra
id_konto	int	NO	PRI	None	auto_increment
nazwa_uzytkownika	varchar(30)	NO		None	
haslo	varchar(30)	NO		None	
email	varchar(50)	NO		None	
id_adres	int	NO	MUL	None	
id_platnosc	int	NO	MUL	None	

tabela **Postać**:

Field	Type	Null	Key	Default	Extra
id_postac	int	NO	PRI	None	auto_increment
nazwa	varchar(30)	NO	UNI	None	
poziom	int	NO		None	
złoto	int	NO		None	
czy_online	tinyint(1)	NO		None	
ostatnio_online	timestamp	YES		None	
wsp_x	int	NO		None	
wsp_y	int	NO		None	
wsp_z	int	NO		None	

tabela **Potwór**:

Field	Type	Null	Key	Default	Extra
id_potwór	int	NO	PRI	None	auto_increment
id_rodzaj	int	NO	MUL	None	
nazwa	varchar(255)	NO		None	
poziom	int	NO		None	
id_łup	int	NO	MUL	None	
wsp_x	int	NO		None	
wsp_y	int	NO		None	
wsp_z	int	NO		None	

### 3.2 Przykłady kilku zapytań i ich wyników

Zapytanie pokazujące ile postaci ma dany użytkownik

```

01 | SELECT Konto.nazwa_uzytkownika, COUNT(Konto.id_konto) AS Postacie
02 | FROM Konto_Postac
03 | INNER JOIN Konto ON Konto.id_konto = Konto_Postac.id_konto
04 | INNER JOIN Postac ON Postac.id_postac = Konto_Postac.id_postac
05 | GROUP BY Konto.nazwa_uzytkownika;

```

Wynik zapytania:

	nazwa_uzytkownika	Postacie
►	ElmilloR	1
	99damage	1
	cherrypach	1
	BlackUFA	2
	Pestily	2
	tsmallbert	1

Zapytanie pokazujące wyposażenie postaci o id = 5 wraz z typem danego wyposażenia i przedmiotu

```

01 | SELECT Postac.nazwa AS 'Postac', Przedmiot.nazwa AS 'Przedmiot',
02 |     Typ_Wyposazenia.typ_wyposazenia AS 'Rodzaj', Typ_Przedmiotu.typ AS '
03 |     Typ'
04 | FROM Wyposazenie_Postaci
05 | INNER JOIN Postac ON Postac.id_postac = Wyposazenie_Postaci.id_postac
06 | INNER JOIN Przedmiot ON Przedmiot.id_przedmiot = Wyposazenie_Postaci.
07 |     id_przedmiot
08 | INNER JOIN Typ_Wyposazenia ON Typ_Wyposazenia.id_typ_wyp =
09 |     Wyposazenie_Postaci.id_typ_wyp
10 | INNER JOIN Typ_Przedmiotu ON Typ_Przedmiotu.id_typ = Przedmiot.id_typ
11 | WHERE Postac.id_postac = 5;

```



Wynik zapytania:

	Postać	Przedmiot	Rodzaj	Typ
▶	Louiso	legendarny hełm	hełm	legendarny
	Louiso	legendarny pancerz	pancerz	legendarny
	Louiso	legendarne buty	buty	legendarny
	Louiso	legendarny miecz	broń	legendarny