

# Projekt bazy - Warhammer Postacie

## 1. Projekt koncepcji, założenia

### 1.1 Zdefiniowanie tematu projektu: krótki opis zadania, zdefiniowanie jego celów i zadań jakie ma realizować:

Temat projektu, na który się zdecydowano jest baza danych postaci graczy do gry RPG Warhammer II Edycja. Pozwala ona na przeglądnięcie aktualnie dostępnych możliwości tworzenia postaci wraz z możliwością wprowadzenia nowego gracza. W jednej aplikacji Mistrz Gry (osoba prowadząca rozgrywkę) może utworzyć postać dla nowego gracza, i przeglądać aktualne możliwości tworzenia oraz parametry strojów, broni, strojów itp.

### 1.2 Analiza wymagań użytkownika: określenie funkcjonalności jakie ma spełniać projektowana baza danych:

Baza ma pozwolić na przeglądanie aktualnej liczby broni, postać klas, itp. jaką gracz może przyjąć oraz aktualizowania liczby graczy poprzez dodanie nowego.

### 1.3 Zaprojektowanie funkcji: określenie podstawowych funkcji realizowanych w bazie danych:

Do funkcji realizowanych w bazie można zaliczyć wyświetlanie statystyk danej rasy i region jej pochodzenia w zależności od wybranej rasy, ekwipunek dla danej profesji i statystyki dla danej klasy wraz wyświetleniem możliwości zostania magiem. Oprócz tego można wprowadzać nowe postaci graczy.

## **2. Projekt diagramów (konceptualny)**

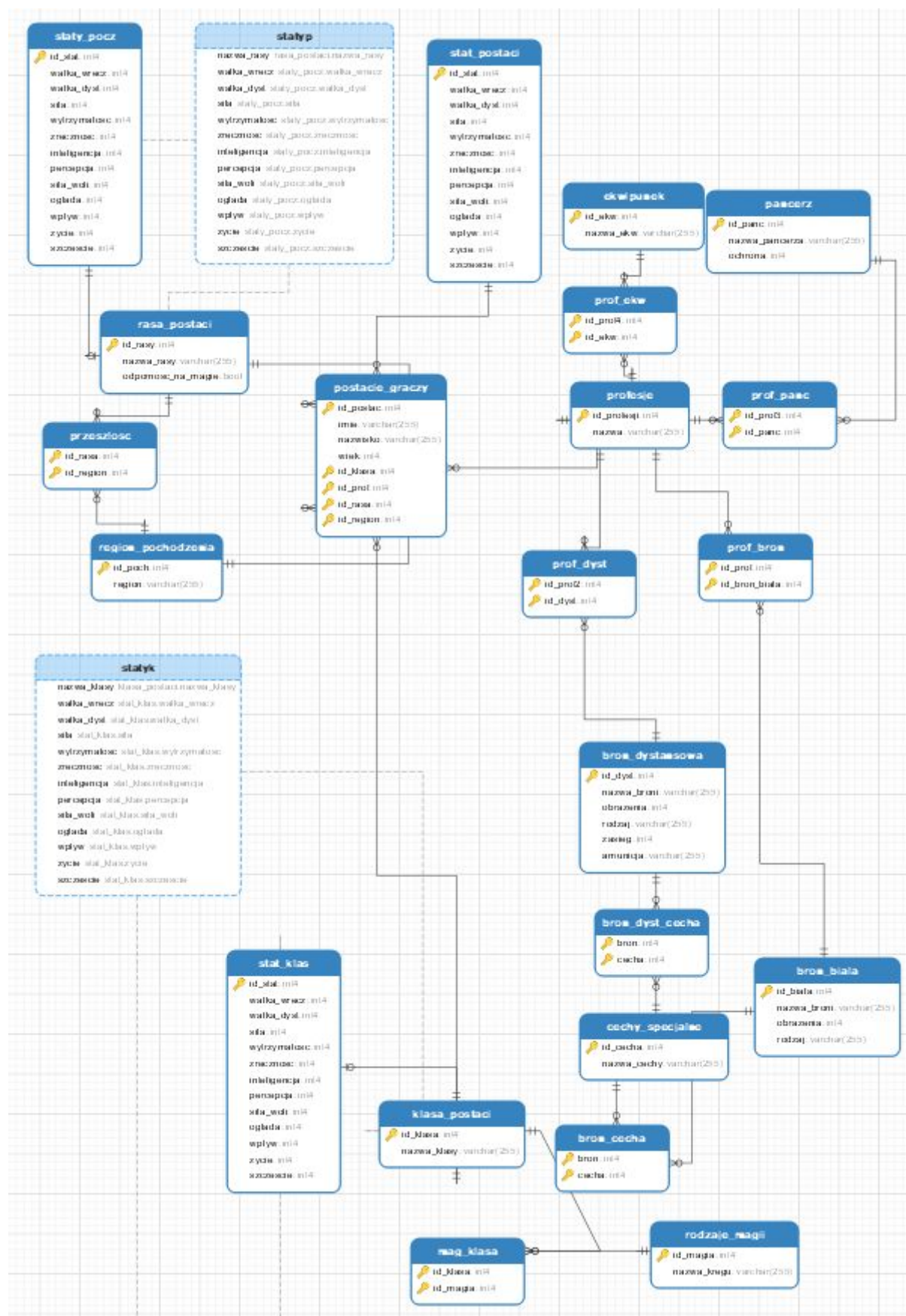
**2.1 Budowa i analiza diagramu przepływu danych (DFD): określenie przepływu danych w systemie (wejścia, wyjścia, operacje, przechowywanie). Należy zastanowić się nad wybraniem elementów, które będą sterowały tym przepływem:**

Wszystkie dane przechowywane są w bazie oraz na bieżąco z niej pobierane a także do niej dodawane za pomocą funkcji i zapytań wykonujących odpowiednie działania bezpośrednio lub na stworzonych wcześniej widokach. Użytkownik korzystając z aplikacji otrzymuje dane w formie tabel pandas DataFrame i przekonwertowane na tabele . Dane wprowadzane przez użytkownika są walidowane w funkcjach a następnie dodawane do bazy.

**2.2 Zdefiniowanie encji (obiektów) oraz ich atrybutów:**

Zdefiniowano encje, atrybuty, klucze własne oraz klucze obce, przedstawione w postaci diagramu ERD w poniższym podpunkcie.

**2.3 Zaprojektowanie relacji pomiędzy encjami: określenia klas encji wraz z atrybutami oraz zdefiniowanie kluczy. Następnie trzeba ustalić powiązania między encjami (eliminacja powiązań wiele-do-wielu). Etap ten powinien prowadzić do ostatecznej definicji logiki bazy danych. Efekt powinien zostać przedstawiony w postaci diagramu ERD:**



Encje są połączone relacjami 1 do 1 lub jeden do wielu.

## **3 Projekt logiczny**

**3.1 Projektowanie tabel, kluczy, indeksów: w oparciu o zdefiniowany diagram ERD należy opisać strukturę bazy danych wraz ze szczegółami technicznymi. Załączyć projekt w języku SQL:**

Bazę danych PostgreSQL w wersji 9.6.13 utworzono w serwisie ElephantSQL. Aplikację napisano w języku Python w wersji 3.8. Użyto do tego bibliotek pythonowych: pandas, psycopg2, dash, dash\_core\_components, dash\_html\_components. Dzięki nim powstał interfejs webowy na porcie 8050. Oprócz tego skorzystano z technologii HTML i CSS. Do utworzenia bazy danych posłużyła nakładka Navicat. Główną tabelą jest tabela postaci graczy, do której są połączone encje za pomocą relacji jeden-do-wielu gdzie są połączone Na podstawie diagramu ERD zaprojektowano relacyjną bazę danych. Zaimplementowano łatwe dodawanie nowych postaci graczy oraz możliwość przeglądania aktualnie dostępnych możliwości. Projekt tabel, połączeń danych został zapisany w pliku proj.sql.

**3.2 Słowniki danych: wypisać słownik danych, określić dziedziny oraz ich ograniczenia:**

Większość danych posiada typ varchar 255 w którym zawarte są opisy, nazwy sprzętu profesji itp. Reszta jest typu int 4 i składają się na niego wszystkie wartości liczbowe oraz ID. Występuje jeden typ boolean określający czy dana rasa jest odporna na magię.

### **3.3 Analiza zależności funkcyjnych i normalizacja tabel (dekompozycja do 3NF ewentualnie BCNF): dokonanie sprawdzenia, czy tabele spełniają warunki zakładanych postaci normalnych ( ew. dekompozycja w celu normalizacji):**

Baza danych jest w postaci 3NF - relacje opisują pojedyncze obiekty (profesje, sprzęt, klasy, magia itp.), wszystkie atrybuty nie posiadają własnej struktury, informacje nie powtarzają się, kolejność wierszy nie ma znaczenia, żadna kolumna niekluczowa nie jest częściowo funkcyjnie zależna od jakiegokolwiek klucza potencjalnego oraz żaden atrybut niekluczowy nie jest zależny funkcyjnie od innych atrybutów niekluczowych. Dwie tabele (staty\_pocz i staty\_klas) są duże i na ich podstawie zrobiono widoki w celu lepszej agregacji danych.

### **3.4. Denormalizacja struktury tabel: ma ona na celu optymalizację przetwarzania, przechowywania danych archiwalnych, dostosowanie do specyficznych wymagań:**

Większość encji ma ustawione jako NOT NULL (nie może być pusta) w celu uniknięcia przypisania roli za pomocą NULL, który często doprowadza do wysypania się programu.

### **3.5. Zaprojektowanie operacji na danych: zdefiniowanie kwerend dla realizacji funkcji wyspecyfikowanych w założeniach projektu (pierwsza część dokumentacji, punkt 3 ). Załączyć projekt w języku SQL:**

Projekt SQL został załączony do projektu wraz z opisem tabel, funkcji, widoków i wyzwalaczem. Funkcje służą do przedstawienia atrybutów, sprzętu w zależności od wyboru rasy, klasy postaci i profesji.

## 4. Projekt funkcjonalny

### 4.1. Interfejsy do prezentacji, edycji i obsługi danych: zdefiniowanie struktury poszczególnych formularzy do wprowadzania danych oraz powiązań między formularzami:

Dane prezentowano formie tabel. Tabele otrzymywało się poprzez wyboru z listy danej wartości i po przyciśnięciu odpowiedniego przycisku otrzymywało się raporty w formie tabel. Poniżej przedstawiono kilka przykładów użycia:

Projekt na przedmiot bazy danych 1 Baza danych graczy do postaci Warhammer	
Autor Jan Drwal	
Wybierz rasę	
WSKAŹ RASĘ	Wybierz rasę i wciśnij przycisk by zobaczyć parametry
Wybierz profesję	
WSKAŹ PROFESJĘ	Wybierz profesję i wciśnij przycisk by zobaczyć parametry
Wybierz klasę	
WSKAŹ KLASĘ POSTACI	Wybierz klasę i wciśnij przycisk by zobaczyć parametry
Poniżej można wprowadzić postać. Parametry dotyczące rasy, klasy i profesji pobierane są z list powyżej	
Wybierz Region	
WPROWADŹ NOWĄ POSTAĆ	imie postaci nazwisko postaci Wybierz dane do list powyżej

Za pomocą 3 dropdownów, 2 text fieldów oraz liczbowej wartości istnieje możliwość wprowadzenia nowej postaci. Poniżej przedstawiono wygląd wprowadzania:

Poniżej można wprowadzić postać. Parametry dotyczące rasy, klasy i profesji pobierane są z list powyżej	
Wybierz Region	
WPROWADŹ NOWĄ POSTAĆ	imie postaci nazwisko postaci Wybierz dane do list powyżej

Przykładowe wyniki po wybraniu odpowiedniej opcji:

Projekt na przedmiot bazy danych 1  
Baza danych graczy do postaci Warhammer

Autor Jan Drwal

Nieumarły

WSKAŹ RASĘ

region	nazwa_rasy	walka_wrecz	walka_dyst	sila	wytrzymalosc	zrecznosc	inteligencja	percepcja	sila_woli	oglada	wplyw	zycie	szczescie
Imperium	nieumarły	20	10	30	30	10	10	20	40	10	5	15	4
Sylvania													
Ulthuan													

grabarz

WSKAŹ PROFESJE

nazwa	obrazenia	cecha	nazwa	ochrona	nazwa	obrazenia	cecha	zasieg	amunicja	nazwa
bron	improwilowane	6	Jednoreczna	ubranie	1	improwilowane	6	duzkie	30	brak
										topota

Tanik

WSKAŹ KLASĘ POSTACI

Klasa nie potrafi używać magii

nazwa_klasy	walka_wrecz	walka_dyst	sila	wytrzymalosc	zrecznosc	inteligencja	percepcja	sila_woli	oglada	wplyw	zycie	szczescie
Tanik	10	0	10	20	0	0	0	10	0	5	5	0

Poniżej można wprowadzić postać. Parametry dotyczące rasy, klasy i profesji pobierane są z list powyżej

Ulthuan

WPROWADŹ NOWĄ POSTAĆ

jan

drwal

56

Wprowadzono nowego gracza

## 4.2 Makropolecenia: powinny ułatwiać obsługę aplikacji z poziomu panelu sterowania oraz ewentualnie z poziomu formularzy:

Zaimplementowano funkcje obsługujące przyciski wraz z dropdownami. Za pomocą przycisków otrzymuje się tabelę a poniżej ich wprowadza się dane, które dodają nową postać oraz aktywują wyzwalacz, aktualizujący tabelę staty postaci.

## 5. Dokumentacja

### 5.1. Wprowadzanie danych: zdefiniowanie sposobu wprowadzania danych ( ręczne, automatyczne, import):

Wszystkie dane zostały do aplikacji wprowadzone ręcznie zgodnie z podręcznikiem do gry pt. “Warhammer 2 edycja gra fabularna”. W razie dodania nowych elementów rozgrywki wystarczy użyć prostego insert into. Nie ma potrzeby aktualizowania danych, gdyż nie ma zmian zasad.

## **5.2 Dokumentacja użytkownika: krótka instrukcja obsługi aplikacji:**

By zobaczyć dane parametru dla rasy, klasy profesji (takie jak sprzęt, broń statystyki itp. ) wybieramy z dropdownów to co nas interesuje i wciskamy odpowiedni przycisk. W celu wprowadzenia nowej postaci wpisujemy odpowiednio imię i nazwisko oraz wybieramy region. POzostałe atrybuty czyli klasa, profesja i rasa wybieramy z powyższych dropdownów. Jeśli wszystko się zgadza dostaniemy informację o wprowadzeniu postaci. Jeśli nie, inforamcja prześle jaki błąd został popełniony (np. wprowadzenie postaci nie magicznej do klasy wymagającej magii).

## **5.3 Opracowanie dokumentacji technicznej: ma to być dokumentacja kodu, która pozwoli na dalszą rozbudowę systemu. W przypadku projektu zrealizowanego w Javie, C++ czy PHP może zostać ona wygenerowana przy użyciu narzędzi automatycznie tworzących dokumentację:**

Dokumentacja techniczna napisana jest w kodzie programu napisanego w Pythonie w formie komentarzy. Polecenia SQL wraz z komentarzami znajdują się w folderze *sql* . W folderze *assets* znajduje się plik CSS do aplikacji Aplikację należy uruchomić poleceniem `python3 tr.py`.

Bibliografia:

- Warhammer II edycja podręcznik główny
- <https://plotly.com/dash/>
- <https://dash.plotly.com/>