



Uniwersytet Gdański
Wydział Matematyki, Fizyki i Informatyki
Instytut Informatyki

The blizzard dismissal

Mateusz Redzimski

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

Gdańsk
24 maja 2020

Spis treści

1	Wprowadzenie	2
2	Opis projektu	3
2.1	Potencjalne grupy użytkowników	4
2.2	Wymagania funkcjonalne	4
2.3	Wymagania niefunkcjonalne	6
2.4	Diagram związków encji	7
3	Przykłady realizacji bazy danych	10
3.1	Przykłady zawartości najważniejszych tabel	10
3.2	Przykłady kilku zapytań i ich wyników	11

1 Wprowadzenie

Baza danych stworzona jest w celu zarządzania wielką ilością danych związaną z nową Grą MMO tego studia. Przechowywane w niej będą informacje na temat założyciela konta, konta, mikro transakcji, postaci stworzonych przez gracza, postaci niezależnych, potworów, przedmiotów, sojuszy, ligi, świata gry oraz ekwipunku postaci.

- System bazodanowy – zbiór przechowywanych w tabelach danych, uzupełniony o informacje w samych tabelach o ściśle określonej strukturze
- Krotka – to wiersz tabeli inaczej zwany REKORDEM
- Tabela – dwuwymiarowa konstrukcja zbudowana z pionowych kolumn i poziomych wierszy, na przecięciu których znajdują się komórki.
- Klucz podstawowy – kolumna z identyfikatorami poszczególnych wierszy.
- Klucz obcy – Kolumna, która zawiera identyfikator rekordu innej tabeli.
- Model relacyjny – model według którego dane przechowywane są w wielu odrębnych, ale powiązanych ze sobą tabelach.
- Postulaty Codda dotyczące struktur danych - Spełnienie postulatów dotyczących struktur danych pozwala w ten sam sposób, niezależnie od wykorzystywanego serwera bazodanowego, zarządzać przechowywanymi w bazach informacjami.
- Postulaty Codda dotyczące przetwarzania danych - Relacyjny model bazy danych został stworzony z myślą o wydajnym i łatwym modyfikowaniu zapisanych w bazach informacji.
- Postulaty Codda dotyczące integralności danych - Przestrzeganie postulatów integralności danych gwarantuje zachowanie logicznej spójności przechowywanych w bazie informacji.
- Normalizacja - procedura wstępnej obróbki danych w celu umożliwienia ich wzajemnego porównywania i dalszej analizy. Przeprowadzana według wyżej wymienionych Postulatów pozwala dążyć kolejno do Pierwszej, Drugiej , Trzeciej oraz Czwartej postaci normalnej.
- Integer– Liczba całkowita o maksymalnej wielkości 2147483647 w programowaniu. Przeznaczone są dla niej 4 bajty pamięci.
- Varchar – to wartość tekstowa zapisana w serwerze bazodanowym. Każdy znak zajmuje 1 bajt a znaków może być od 0 do 8000.
- Postać niezależna – Postać sterowana przez program a nie gracza.
- Sklep premiowany – Płatności w tym sklepie przebiegają przy użyciu waluty rzeczywistej.
- Relacje - (związki) to zależności między tabelami umożliwiające ich logiczne powiązanie ze sobą. Dzielimy na :

- Jeden-do-jednego - polega na tym, że jednemu rekordowi pierwszej tabeli jest przyporządkowany dokładnie jeden rekord drugiej tabeli, a jednemu rekordowi drugiej tabeli jest przyporządkowany dokładnie jeden rekord pierwszej tabeli.
- Jeden-do-wielu - polega na tym, że jednemu rekordowi pierwszej tabeli jest przyporządkowanych wiele rekordów drugiej tabeli, a jednemu rekordowi drugiej tabeli jest przyporządkowany dokładnie jeden rekord pierwszej tabeli,
- Wiele-do-wielu polega na tym, że jednemu rekordowi pierwszej tabeli jest przyporządkowanych wiele rekordów drugiej tabeli, a jednemu rekordowi drugiej tabeli jest przyporządkowanych wiele rekordów pierwszej tabeli.

2 Opis projektu

Projekt „The blizzard dismissal” jest odpowiedzią na masowe zwolnienia w firmie „Blizzard Entertainment”. W firmie tej pojawiło się wolne miejsce pracy na stanowisku „Programista baz danych”. Założeniem projektu było dostarczenie do osoby rekrutującej zaprojektowanej bazy danych według własnego uznania. Zadanie to miało na celu sprawdzić nie tylko to jak dany pracownik odnajdywał by się w roli Programisty baz danych ale również jego umiejętności kreatywnego myślenia i dostosowania się do sytuacji. Aspirujący do tego stanowiska kandydat dostał do swojej dyspozycji podstawowe, przykładowe założenia towarzyszące tego rodzaju bazie danych typu:

- Dane graczy
- Postacie
- Przedmioty
- Mikro transakcje
- Sojusze
- Ligi
- Lokalizacja postaci w świecie gry
- Statystyki postaci
- Punkty uzyskane przez postać lub sojusz
- Historia Mikro transakcji
- Ekwipunek postaci

Resztę zagadnień oraz wizję gry, kandydat miał zrealizować samemu. To jak poradzi sobie z zadaniem oraz jak wiele rzeczy wykreuje, pokaże czy praca w zespole „Blizzard Entertainment” stoi dla niego otworem. Zadanie to w pełni pokazuje jak dana osoba radzi sobie z wszystkimi zagadnieniami związanymi z tego rodzaju rzemiosłem.

2.1 Potencjalne grupy użytkowników

- Administrator – główny zarządca bazy danych, posiada pełen dostęp do bazy danych
- Moderator – osoba której zadaniem jest nadzorowanie prac związanych z bazą danych. Jej dostęp jest ograniczony w stosunku do Administratora o informacje których Administrator sobie zażyczy.
- Gracz pełnoletni – osoba która korzysta z baz danych w celu rozgrywki. Jej dostęp do bazy danych jest ograniczony w stosunku do Moderadora o informacje których Administrator sobie zażyczy.
- Gracz niepełnoletni – osoba która korzysta z baz danych w celu rozgrywki . Jej dostęp do bazy danych jest ograniczony w stosunku do Gracza pełnoletniego, o informacje których Administrator sobie zażyczy, po uzgodnieniu z Rodzicem bądź Opiekunem prawnym gracza.

2.2 Wymagania funkcjonalne

Informacje przechowywane w bazie danych na temat :

- Założyciela konta oraz konta: Adres zamieszkania, data urodzenia, login, hasło, stan portfela premiowanego, historia mikro transakcji, serwera który wybrał, koszyka z sklepu premiowanego, język jakim się posługuje.
- Mikro transakcji: przedmioty premiowane oraz ich cena.
- Postaci stworzonych przez gracza: W jakiej lidze grają, w jakim świecie się znajdują, ich położenie w świecie gry, nazwa, statystyki, sojusz do którego przynależą, doświadczenie przez nie zebrane, poziom, wskaźnik honoru, ekwipunku , profesja, klasa, rasa, płeć, strona konfliktu oraz ogólny wynik postaci.
- Postaci niezależnych: Nazwa, poziom, rasa, zdobyte doświadczenie, misji, ekwipunku, statystyki oraz położenie w świecie gry.
- Potworów: Statystyki, przedmioty które z nich wypadają, położenie w świecie gry, poziom unikalności, typ, poziom oraz nazwa.
- Przedmiotów: Statystyki, nazwa, unikalność, szansa na zdobycie, informacja o tym skąd można przedmiot zdobyć oraz poziom przedmiotu.
- Sojuszy: Nazwa, maksymalna ilość członków, aktualna ilość członków, serwer na którym się znajduje oraz ogólnego wyniku w rankingu.
- Ligi: Data rozpoczęcia, data zakończenia, nazwa.
- Ekwipunku postaci: Przedmioty jakie zawiera, do jakiej postaci przynależy.

Zadania do jakich użyta zostanie baza danych:

- Stworzenia konta oraz możliwość zalogowania na konto.

- Dodawania oraz usuwania przedmiotów premiowanych do gry.
- Kupno i sprzedaż przedmiotów premiowanych.
- Doładowywania swojego portfela premiowanego walutą ustaloną przez Administratora.
- Stworzenie postaci oraz możliwość jej usunięcia.
- Sprawdzenie statystyk postaci oraz ich rozwój.
- Zmienienie serwera gry, nazwy postaci, klasy, profesji oraz rasy .
- Stworzenie rankingu graczy/sojuszy opartego na ich wyniku ogólnym.
- Wprowadzania lig oraz grania na nich.
- Usuwanie lig.
- Stworzenie sojuszu oraz możliwość jego usunięcia.
- Dołączenie postaci do sojuszu oraz możliwość odejścia z niego.
- Możliwość przekazania dowództwa nad sojuszem lub usunięcia postaci z sojuszu.
- Dodawanie oraz usuwanie postaci niezależnych/potworów do gry.
- Wczytywanie informacji o ilości osób grających na serwerze/mapie oraz ich położeniu.
- Wyświetlanie informacji na temat potwora/postaci niezależnej/postaci gracza.
- Tworzenie przedmiotów oraz ich usuwanie.
- Wyświetlanie/zmiana dostępnych przedmiotów oraz ich statystyk.
- Handel przedmiotami.
- Zdobywanie przedmiotów poprzez walkę z potworami.
- Zmiana języka gry którym się posługujemy.
- Wyświetlenie listy potworów dostępnych w danej lokacji.
- Wyświetlenie przedmiotów możliwych do zdobycia z danego potwora.
- Wyświetlenie listy zadań do wykonania od danej postaci niezależnej.
- Możliwość poruszania się po świecie gry oraz korzystania z mapy.
- Możliwość rozmowy z postacią niezależną.
- Możliwość rozmowy między graczami.
- Analiza aktywności graczy na podstawie danych.
- Analiza rozłożenia demograficznego graczy.

- Analiza rynku na którym gra zarabia najwięcej.
- Możliwość założenia oraz zdjęcia przedmiotu.
- Możliwość zdobycia lub wyrzucenia przedmiotu.
- Możliwość użycia przedmiotu.

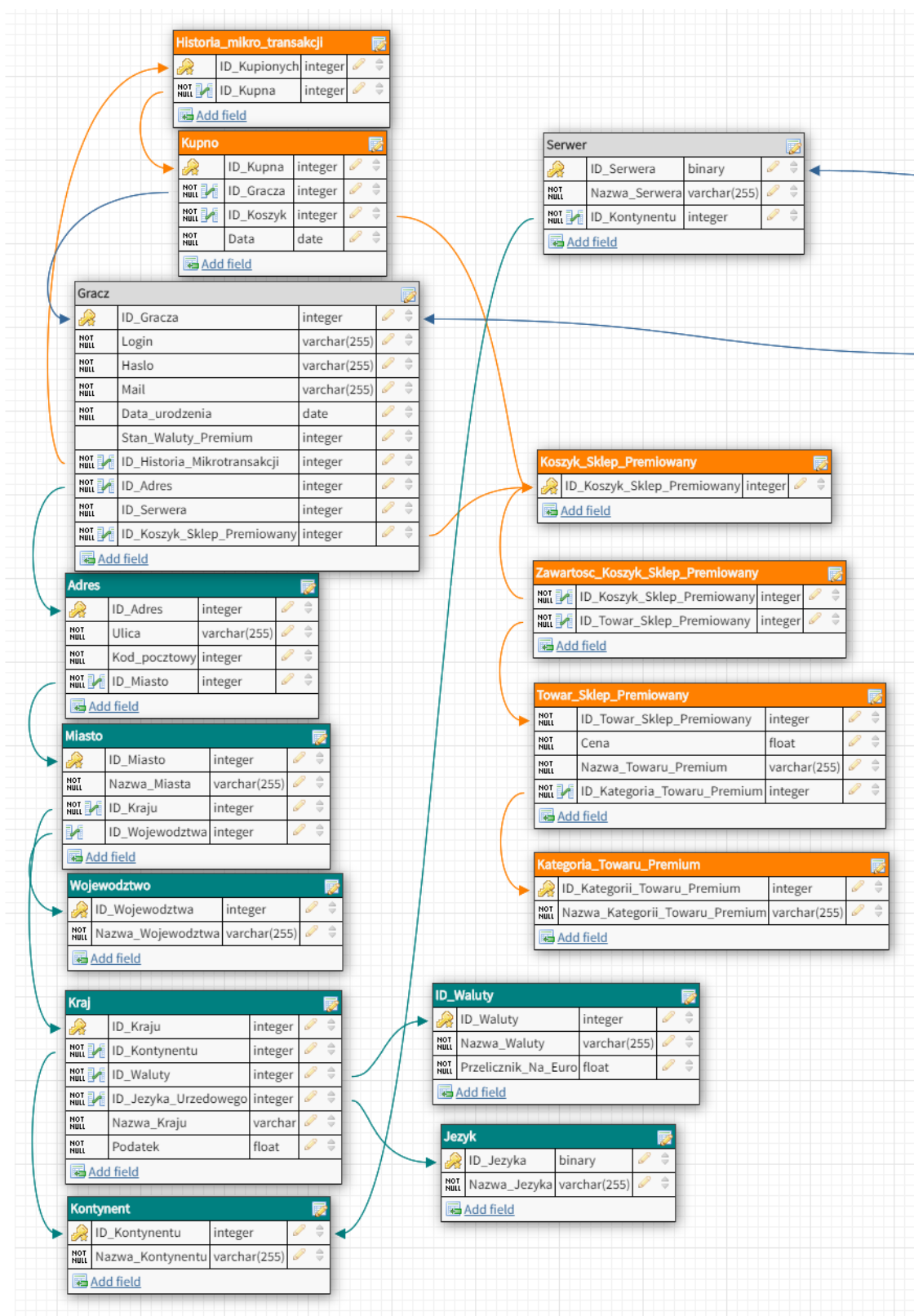
2.3 Wymagania niefunkcjonalne

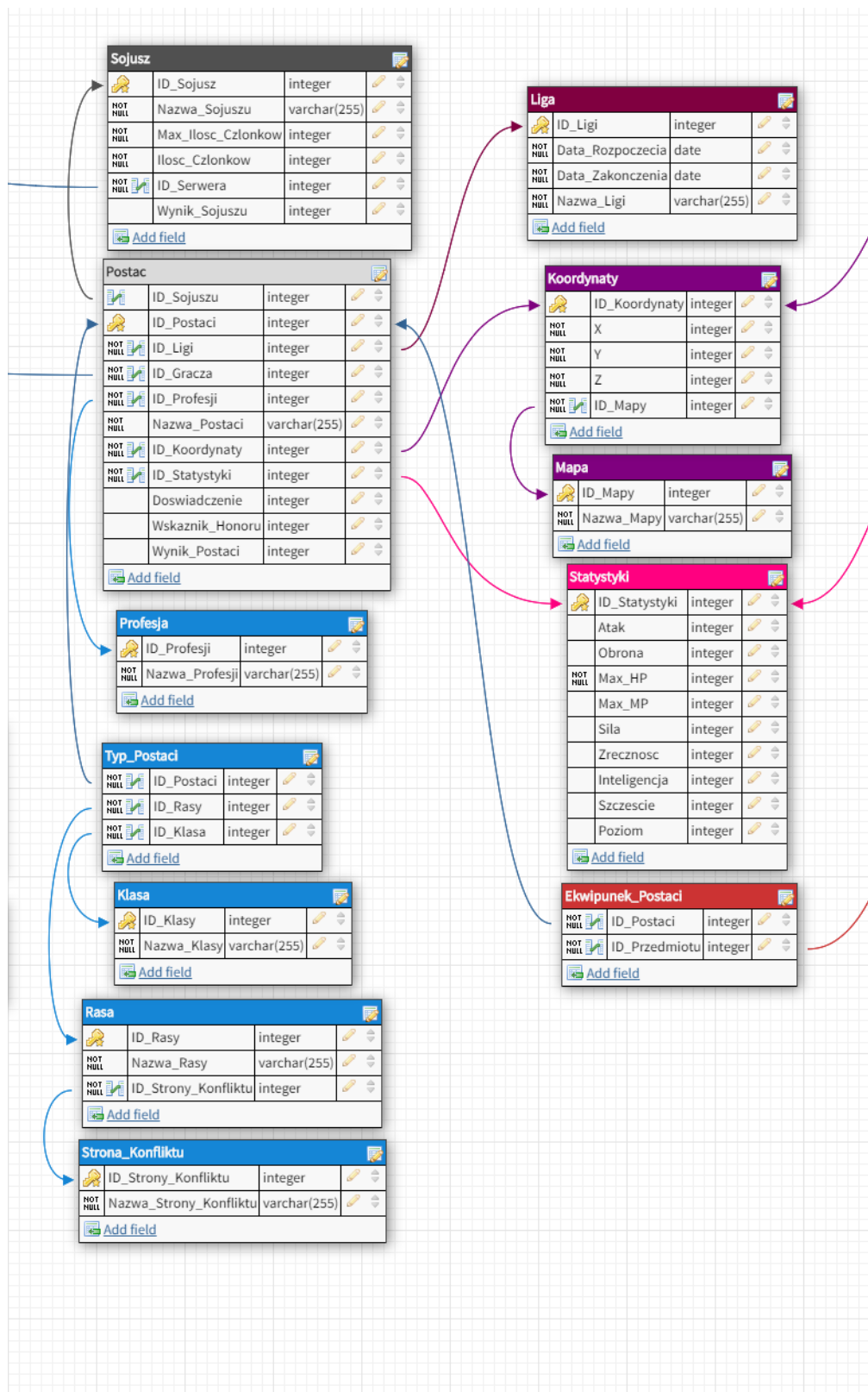
Baza danych jest zrealizowana na podstawie modelu relacyjnego. System zarządzania użyty do stworzenia tej bazy danych to [1] POSTGRESQL w wersji 12.3. Do stworzenia tej bazy danych potrzebny był schemat ERD który stworzyłem przy użyciu programu DBDESIGNER. Jego zaletami jest prostota tworzenia tabel i dalszej ich edycji oraz przejrzystość w wykonywanych zadaniach. Największym minusem tego programu jest jego cena oraz ograniczenia. W wersji basic za którą przyszło mi zapłacić 7,99\$ za miesiąc użytkowania mogłem stworzyć jedynie 30 tabel. Przy większych projektach takie ograniczenia uniemożliwią dalszą pracę.

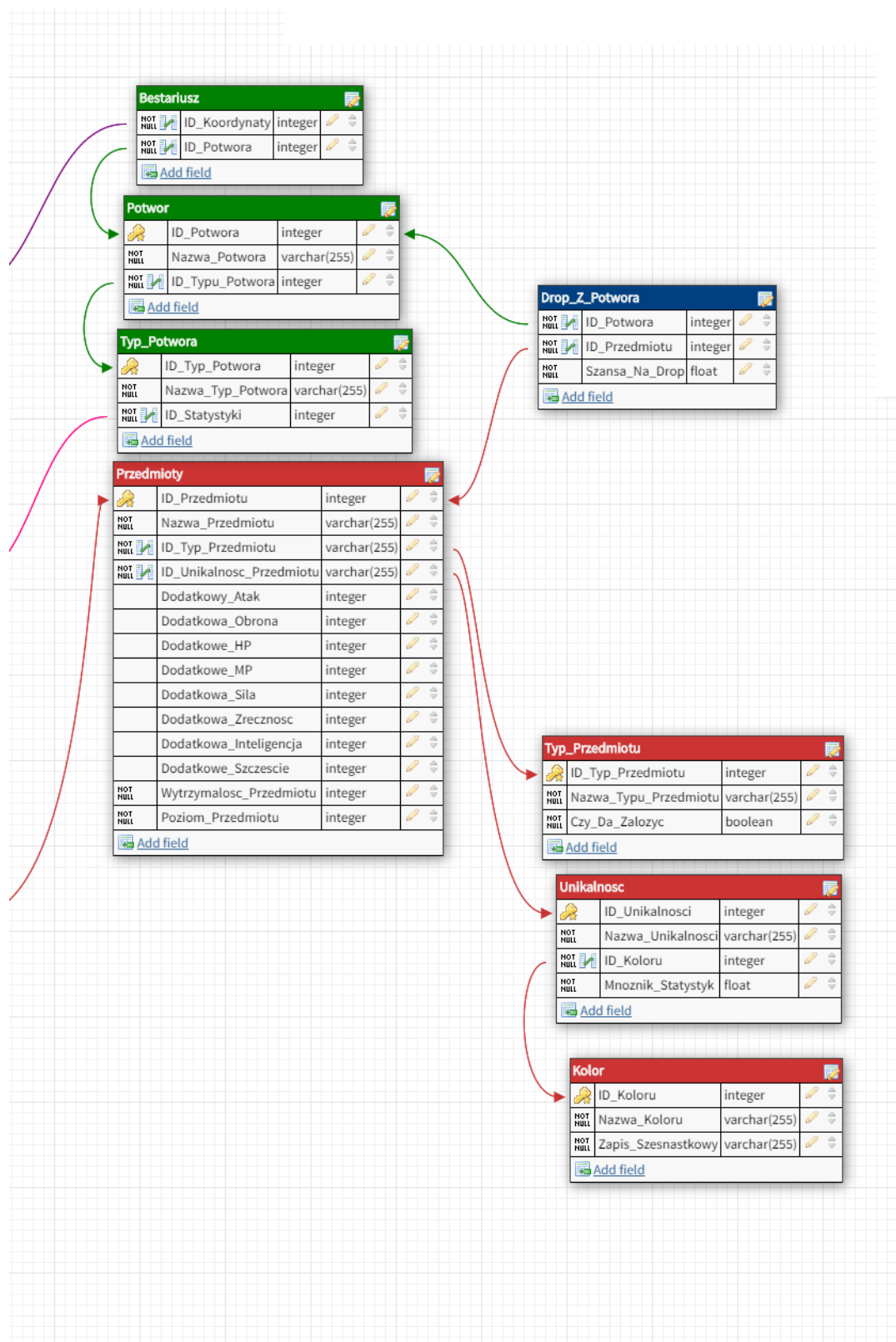
System zarządzania POSTGRESQL ma wiele zalet, między innymi fakt, że jest to darmowy program opierający się na licencji open source. Ma silną społeczność przez co łatwo znaleźć odpowiedź na nasze pytanie lub uzyskać wsparcie. Wadą POSTGRESQL jest natomiast wydajność która może wydawać się gorsza od jego odpowiedników np. MySQL.

Do wprowadzenia zewnętrznego serwera bazodanowego zdecydowałem się na platformie AZURE która zaintrygowała mnie darmowym okresem próbnym oraz niskim progiem wejścia. Pozwoliło to na szybkie oswojenie się z podstawowymi funkcjami platformy.

2.4 Diagram związków encji







3 Przykłady realizacji bazy danych

Przykłady zawartości najważniejszych tabel są pokazane w formacie MySQL. Przykłady kilku zapytań i ich wyników są zapisane w formacie Postgresql.

3.1 Przykłady zawartości najważniejszych tabel

TABELA GRACZ.

Field	Type	Null	Key	Default	Extra
ID_Gracza	int	NO	PRI	NULL	auto_increment
Login	varchar(255)	NO		NULL	
Haslo	varchar(255)	NO		NULL	
Mail	varchar(255)	NO		NULL	
Data_urodzenia	date	NO		NULL	
Stan_Waluty_Premium	int	YES		NULL	
ID_Historia_Mikrotransakcji	int	NO		NULL	
ID_Adres	int	NO		NULL	
ID_Serwera	int	NO		NULL	
ID_Koszyk_Sklep_Premiowany	int	NO		NULL	

TABELA POSTAĆ.

Field	Type	Null	Key	Default	Extra
ID_Sojuszu	int	YES		NULL	
ID_Postaci	int	NO	PRI	NULL	auto_increment
ID_Ligi	int	NO		NULL	
ID_Gracza	int	NO		NULL	
ID_Profesji	int	NO		NULL	
Nazwa_Postaci	varchar(255)	NO		NULL	
ID_Koordynaty	int	NO		NULL	
ID_Statystyki	int	NO		NULL	
Doswiadczenie	int	YES		NULL	
Wskaznik_Honoru	int	YES		NULL	
Wynik_Postaci	int	YES		NULL	

TABELA PRZEDMIOTY

Field	Type	Null	Key	Default	Extra
ID_Przedmiotu	int	NO	PRI	NULL	auto_increment
Nazwa_Przedmiotu	varchar(255)	NO		NULL	
ID_Typ_Przedmiotu	int	NO		NULL	
ID_Unikalnosc_Przedmiotu	int	NO		NULL	
Dodatkowy_Atak	int	YES		NULL	
Dodatkowa_Obrona	int	YES		NULL	
Dodatkowe_HP	int	YES		NULL	
Dodatkowe_MP	int	YES		NULL	
Dodatkowa_Sila	int	YES		NULL	
Dodatkowa_Zrecznosc	int	YES		NULL	
Dodatkowa_Inteligencja	int	YES		NULL	
Dodatkowe_Szczescie	int	YES		NULL	
Wytrzymalosc_Przedmiotu	int	YES		NULL	
Poziom_Przedmiotu	int	YES		NULL	

3.2 Przykłady kilku zapytań i ich wyników

Utworzenie tabeli postaci.

```

01 | CREATE TABLE "Postac" (
02 |     "ID_Sojuszu" integer,
03 |     "ID_Postaci" serial NOT NULL,
04 |     "ID_Ligi" integer NOT NULL,
05 |     "ID_Gracza" integer NOT NULL,
06 |     "ID_Profesji" integer NOT NULL,
07 |     "Nazwa_Postaci" varchar(255) NOT NULL,
08 |     "ID_Koordynaty" integer NOT NULL,
09 |     "ID_Statystyki" integer NOT NULL,
10 |     "Doswiadczenie" integer,
11 |     "Wskaznik_Honoru" integer,
12 |     "Wynik_Postaci" integer,
13 |     CONSTRAINT "Postac_pk" PRIMARY KEY ("ID_Postaci")
14 | ) WITH (
15 |     OIDS=FALSE
16 | );

```

Utworzenie tabeli przedmiotów.

```

01 | CREATE TABLE "Przedmioty" (
02 |     "ID_Przedmiotu" serial NOT NULL,
03 |     "Nazwa_Przedmiotu" varchar(255) NOT NULL,
04 |     "ID_Typ_Przedmiotu" integer NOT NULL,
05 |     "ID_Unikalnosc_Przedmiotu" integer NOT NULL,
06 |     "Dodatkowy_Atak" integer,
07 |     "Dodatkowa_Obrona" integer,
08 |     "Dodatkowe_HP" integer,

```

```
09 |         "Dodatkowe_MP" integer,  
10 |         "Dodatkowa_Sila" integer,  
11 |         "Dodatkowa_Zrecznosc" integer,  
12 |         "Dodatkowa_Inteligencja" integer,  
13 |         "Dodatkowe_Szczescie" integer,  
14 |         "Wytrzymalosc_Przedmiotu" integer,  
15 |         "Poziom_Przedmiotu" integer,  
16 |         CONSTRAINT "Przedmioty_pk" PRIMARY KEY ("ID_Przedmiotu")  
17 | ) WITH (  
18 |     OIDS=FALSE  
19 | );
```

Utworzenie tabeli Ekwipunek Postaci.

```
01 | CREATE TABLE "Ekwipunek_Postaci" (  
02 |     "ID_Postaci" integer NOT NULL,  
03 |     "ID_Przedmiotu" integer NOT NULL  
04 | ) WITH (  
05 |     OIDS=FALSE  
06 | );
```

Relacja między Postacią a Statystykami.(relacja jeden do wielu)

```
01 | ALTER TABLE "Postac" ADD CONSTRAINT "Postac_fk5" FOREIGN KEY ("ID_Statystyki"  
    ") REFERENCES "Statystyki"("ID_Statystyki");
```

Relacja między Postacią a Przedmiotami za pośrednictwem tabeli Ekwipunku Postaci.
(relacja wiele do wielu)

```
01 | ALTER TABLE "Ekwipunek_Postaci" ADD CONSTRAINT "Ekwipunek_Postaci_fk0"  
    FOREIGN KEY ("ID_Postaci") REFERENCES "Postac"("ID_Postaci");  
02 | ALTER TABLE "Ekwipunek_Postaci" ADD CONSTRAINT "Ekwipunek_Postaci_fk1"  
    FOREIGN KEY ("ID_Przedmiotu") REFERENCES "Przedmioty"("ID_Przedmiotu");
```

Wstawienie rekordu do tabeli gracz.

```
01 | INSERT INTO "Gracz" VALUES(1,'Frevoste','Fajnehaslo123#','Fajnymail123@ug.  
    edu.pl','2017-03-14',1,1,1,1,1);
```

Usunięcie rekordu z LOGINEM FREVOSTE z tabeli gracz.

```
01 | DELETE FROM "Gracz" WHERE "Login"='Frevoste';
```

Wyświetlenie kolumny ID Gracza z tabeli Gracz.

```
01 | SELECT "ID_Gracza" from "Gracz";
```

Wyświetlenie ekwipunku postaci z ID Postaci=1.

```
01 | SELECT * FROM "Ekwipunek_Postaci"  
02 | WHERE "ID_Postaci"=1;
```

Wykaz statystyk postaci o ID Postaci =1.

```
01 | SELECT * FROM "Statystyki"
02 | AS P
03 | WHERE P."ID_Statystyki"
04 | IN (SELECT K."ID_Statystyki"
05 | FROM "Postac" as K
06 | WHERE "ID_Postaci" = 1);
```

Wykaz postaci na mapie o ID Mapy = 1.

```
01 | SELECT "ID_Postaci"
02 | FROM "Postac"
03 | AS P
04 | WHERE P."ID_Koordynaty"
05 | IN (SELECT K."ID_Koordynaty"
06 | FROM "Koordynaty" as K
07 | WHERE "ID_Mapy" = 1);
```

Wykaz punktów uzyskanych przez postać z ID Postaci=1.

```
01 | SELECT "Wynik_Postaci" FROM "Postac" WHERE "ID_Postaci"=1;
```

Ranking postaci na podstawie uzyskanych punktów.

```
01 | SELECT "Wynik_Postaci","ID_Postaci","Nazwa_Postaci"
02 | FROM "Postac"
03 | ORDER BY "Wynik_Postaci";
```

Ranking sojuszy na podstawie uzyskanych punktów.

```
01 | SELECT "Wynik_Sojuszu","ID_Sojuszu","Nazwa_Sojuszu"
02 | FROM "Sojusz"
03 | ORDER BY "Wynik_Sojuszu";
```

Historia mikro transakcji Gracza o ID Gracza=1.

```
01 | SELECT * FROM "Historia_mikro_transakcji"
02 | AS P
03 | WHERE P."ID_Kupionych"
04 | IN (SELECT K."ID_Historia_Mikrotransakcji"
05 | FROM "Gracz" as K
06 | WHERE "ID_Gracza" = 1);
```

Literatura

- [1] The PostgreSQL Global Development Group, *Oficjalna dokumentacja postgresql*, 2020.