

Dokumentacja bazy danych gry komputerowej

Opis projektu

Baza danych jednoosobowej gry komputerowej typu RPG powiązuje postać z użytkownikiem, klasą, miejscem, zadaniami, potworami oraz przedmiotami.

Najważniejsze encje i ich atrybuty

Encja **uzytkownik** przechowuje dane logowania użytkowników:

- **login** (varchar (20)) : klucz główny encji, wprowadzany przez użytkownika, może zawierać do 20 znaków,
- **haslo** (varchar (32)) : wprowadzane przez użytkownika, może zawierać do 32 znaków, nie dopuszcza wartości NULL.

Encja **klasa** przechowuje dostępne klasy postaci w grze:

- **nazwa** (varchar (20)) : nazwa klasy, klucz główny encji,
- **opis** (text) : opis klasy, dopuszcza wartość NULL,
- **sila, wytrzymalosc, zrecznosc, inteligencja, charyzma, szczescie** (int) : statystyki klasy, każda z nich jest osobnym atrybutem, którego domyślna wartość to 10.

Encja **miejsce** jest tabelą słownikową, która przechowuje nazwy miejsc dostępnych w grze:

- **nazwa** (varchar (32)) : nazwa miejsca, klucz główny encji.

Encja **zadanie** przechowuje dostępne zadania w grze:

- **id** (serial) : automatycznie generowany klucz główny encji,
- **nazwa** (varchar (50)) : unikalna nazwa zadania, nie dopuszcza wartości NULL,
- **miejsce** (varchar (20)) : określa w miejsce, w którym można wykonać zadanie, dopuszcza wartość NULL, co oznacza, że zadanie można wykonać w dowolnym miejscu,
- **pd** (int) : liczba punktów doświadczenia za wykonanie zadania, nie dopuszcza wartości NULL,
- **pieniadze** (int) : ilość pieniędzy za wykonanie zadania, nie dopuszcza wartości NULL,
- **opis** (text) : opis zadania, dopuszcza wartość NULL.

Encja **potwor** przechowuje istniejące potwory w grze:

- **id** (serial) : automatycznie generowany klucz główny encji,
- **nazwa** (varchar (20)) : unikalna nazwa potwora, nie dopuszcza wartości NULL,
- **atak** (int) : punkty ataku potwora, nie dopuszcza wartości NULL,
- **zdrowie** (int) : punkty zdrowia potwora, nie dopuszcza wartości NULL.

Encja **postac** przechowuje postacie stworzone przez użytkowników w grze:

- **id** (serial) : automatycznie generowany klucz główny encji,
- **nazwa** (varchar (20)) : nazwa postaci wprowadzana przez użytkownika, może zawierać do 20 znaków, nie dopuszcza wartości NULL,
- **uzytkownik** (varchar (20)) : login użytkownika przypisanego do postaci, jest to wartość unikalna, ponieważ użytkownik może mieć przypisaną maksymalnie jedną postać, nie dopuszcza wartości NULL,
- **status** (boolean) : informacja, czy postać żyje (TRUE), czy jest martwa (FALSE),
- **klasa** (varchar (20)) : klasa przypisana do postaci, klucz obcy z encji **klasa**, nie dopuszcza wartości NULL,

- **miejsce** (varchar (20)) : miejsce, w którym aktualnie przebywa postać, klucz obcy z encji miejsce, nie dopuszcza wartości NULL.

Encja **typ_przedmiotu** jest tabelą słownikową, która przechowuje nazwy typów przedmiotów:

- **nazwa** (varchar (32)) : nazwa typu przedmiotu, klucz główny encji.

Encja **przedmiot** przechowuje istniejące przedmioty w grze:

- **id** (serial) : automatycznie generowany klucz główny encji,
- **nazwa** (varchar (20)) : unikalna nazwa przedmiotu, nie dopuszcza wartości NULL,
- **typ** (varchar (20)) : określa typ przedmiotu klucz obcy z encji typ_przedmiotu, nie dopuszcza wartości NULL,
- **opis** (text) : opis przedmiotu, dopuszcza wartość NULL,
- **sila, wytrzymalosc, zrecznosc, inteligencja, charyzma, szczescie** (int) : statystyki przedmiotu, każda z nich jest osobnym atrybutem, którego domyślna wartość to 0.

Związki między encjami

Związek **jednojednoznaczny**:

- **uzytkownik - postac**: jeden użytkownik może posiadać maksymalnie jedną postać (0:1), do postaci jest przypisany jeden konkretny użytkownik (1:1).

Związki **jednoznaczne**:

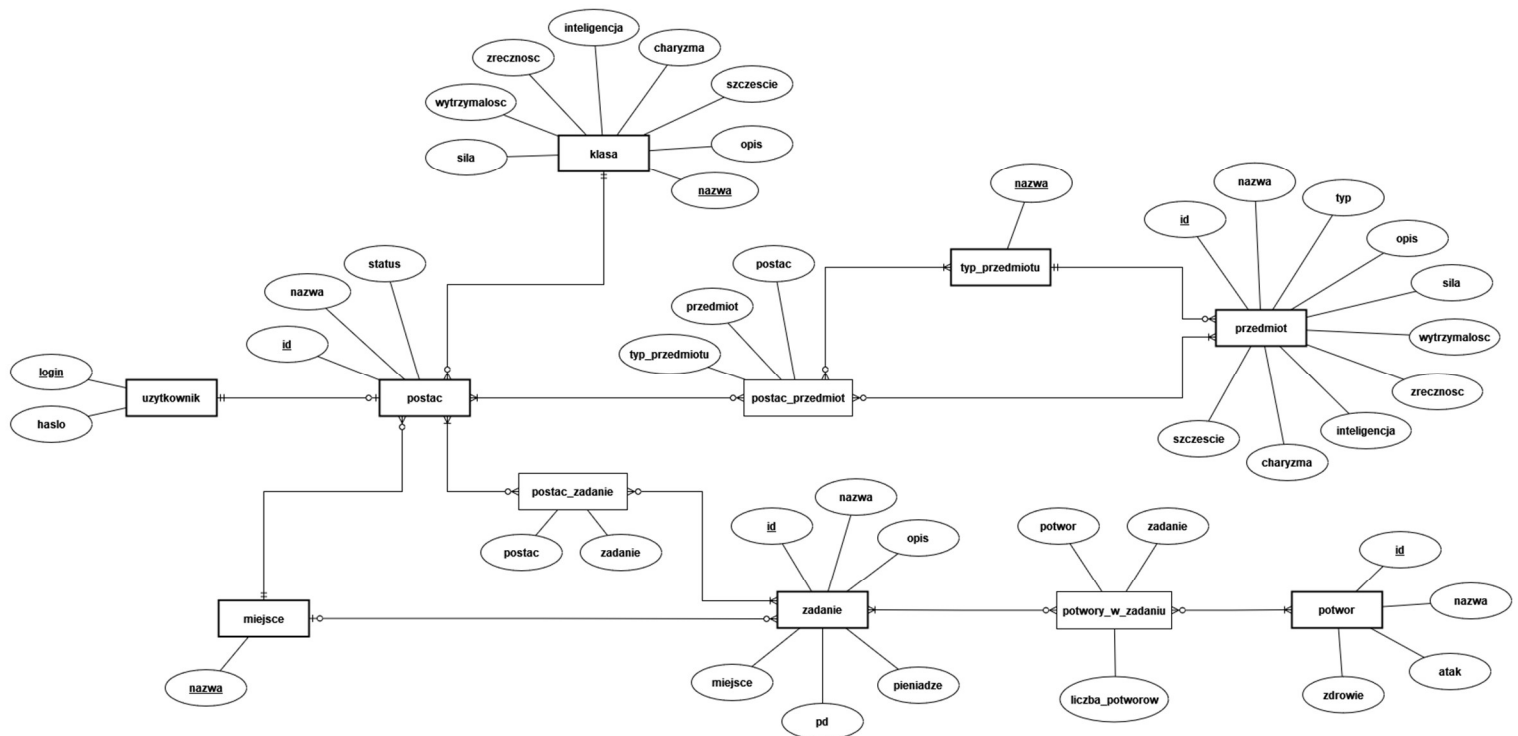
- **klasa - postac**: klasa może być przypisana do wielu postaci (0:∞), ale każda postać musi mieć przypisaną jedną konkretną klasę (1:1),
- **miejsce - zadanie**: miejsce może być przypisane do wielu zadań (0:∞), ale każde zadanie może być przypisane do maksymalnie jednego miejsca (0:1),
- **miejsce - postac**: miejsce może być przypisane do wielu postaci (0:∞), ale każda postać musi przebywać w jednym konkretnym miejscu (1:1),
- **przedmiot - typ_przedmiotu**: przedmiot ma przypisany jeden konkretny typ przedmiotu (1:1), a typ przedmiotu może być przypisany do wielu przedmiotów (0:∞).

Związki **wieloznaczne**:

- **postac - zadanie przez postac_zadanie**: postać może mieć wykonane wiele zadań (0:∞), a każde zadanie mogło zostać wykonane przez różne postacie (0:∞), ale postać może wykonać dane zadanie tylko jeden raz, co sprawdza reguła UNIQUE (postac, zadanie),
- **potwor - zadanie przez potwory_w_zadaniu**: potwór może występować w wielu zadaniach (0:∞), a każde zadanie może dotyczyć wielu potworów (0:∞), ale dany potwór może być przypisany do danego zadania tylko jeden raz, co sprawdza reguła UNIQUE(zadanie, potwor),
- **postac - przedmiot - typ_przedmiotu przez postac_przedmiot**: postać może w danej chwili używać wielu przedmiotów (0:∞), a każdy przedmiot może być używany przez wiele postaci jednocześnie (0:∞), jednak postać może posiadać tylko jeden przedmiot danego typu (0:1), co sprawdza reguła UNIQUE (postac, typ_przedmiotu) .

Przewidywane pytania

1. Wyświetlenie opisów wszystkich zadań, których nazwa nie zaczyna się od 'Ochrona' lub 'Ratowanie'.
2. Wyświetlenie nazw i opisów klas, które nie zostały przypisane do żadnej postaci.
3. Wyświetlenie przypisania potworów do zadań, ale tylko te rekordy, w których liczba potworów jest w przedziale od 2 do 8, oraz, w których nazwa potwora nie składa się z dokładnie trzech liter.



Fragmenty tabel

tabela uzytkownik:

login	haslo
adam	haslo123
janek23	42jhqla3618

tabela klasa:

nazwa	opis	sila	wytrzymalosc	zrecznosc	inteligencja	charyzma	szczęście
barbarzyńca	Zaciekły wojownik potrafiący wpaść w szal bitewny.	65	50	10	5	0	20
bard	Inspirujący czarodziej, którego moc rezonuje z muzyką.	10	50	20	30	25	15

tabela miejsce:

nazwa
miasto
las

tabela zadanie:

id	nazwa	miejsce	pd	pieniadze	opis
1	Zaginiony podróżnik	las	150	80	Odszukaj podróżnika, który zaginął w lesie.
2	Rzadkie zioła		200	100	Zbierz zioła, które rosną w różnych częściach mapy.
3	Ochrona konwoju	miasto	280	200	Ochroń konwój handlowy przed trolami i goblinami.

tabela potwor:

id	nazwa	atak	zdrowie
1	Goblin	10	50
2	Wilk	20	60
3	Ork	12	70

tabela postac:

id	nazwa	uzytkownik	status	klasa	miejsce
1	Asmodeusz	_adam_	t	łotr	tawerna
2	Gajusz	janek23	t	druid	las
3	Hypatia	kasja	f	druid	ruiny

tabela przedmiot:

id	nazwa	typ	opis	sila	wytrzymalosc	zrecznosc	inteligencja	charyzma	szczęście
1	Miecz	broń	Standardowy miecz dla wojownika.	10	0	2	0	0	0
2	Księga zaklęć	narzędzie	Księga zawierająca zaklęcia dla maga.	0	0	0	12	0	0

tabela potwory_w_zadaniu:

zadanie	potwor	liczba_potworow
1	2	3
2	4	6

tabela postac_zadanie:

postac	zadanie
1	3
1	6

tabela postac_przedmiot:

postac	przedmiot	typ_przedmiotu
1	1	broń
1	8	tarcza
2	2	narzędzie

Przykłady zapytań

Wyświetlenie nazw i opisów klas, które nie zostały przypisane do żadnej postaci.

```
SELECT nazwa, opis
FROM klasa k
WHERE NOT EXISTS
    (select *
     FROM postac p
     WHERE p.klasa = k.nazwa)
ORDER BY k.nazwa;
```

Wyświetlenie przypisania potworów do zadań, ale tylko te rekordy, w których liczba potworów jest w przedziale od 2 do 8, oraz, w których nazwa potwora nie ma trzech liter.

```
SELECT z.nazwa AS zadanie, p.nazwa AS potwor, pwz.liczba_potworow
FROM potwory_w_zadaniu pwz, zadanie z, potwor p
WHERE pwz.zadanie = z.id
    AND pwz.potwor = p.id
    AND p.nazwa NOT LIKE '___'
    AND pwz.liczba_potworow BETWEEN 2 AND 8
ORDER BY z.id, liczba_potworow;
```

Wyświetlenie przypisania przedmiotów do postaci, które aktualnie używają więcej niż jednego przedmiotu

```
SELECT postac.nazwa AS postac, przedmiot.nazwa AS przedmiot
FROM (postac_przedmiot INNER JOIN postac ON postac_przedmiot.postac = postac.id)
    INNER JOIN przedmiot ON postac_przedmiot.przedmiot = przedmiot.id
WHERE postac IN
    (SELECT postac
     FROM postac_przedmiot
     GROUP BY postac
     HAVING COUNT(postac) > 1);
```

Stworzenie widoku: wyświetlenie wszystkich informacji o postaciach, obliczenie ich punktów doświadczenia, pieniędzy oraz poziomu na podstawie wykonanych zadań.

```
CREATE VIEW postac_informacje AS
SELECT p.*,
    COALESCE(suma.pd, 0) AS pd,
    COALESCE(suma.pieniadze, 0) AS pieniadze,
    COALESCE(suma.pd/120, 0) AS poziom
FROM postac p
LEFT JOIN
    (SELECT pz.postac,
        SUM(zadanie.pd) AS pd,
        SUM(zadanie.pieniadze) AS pieniadze
     FROM postac_zadanie pz
     INNER JOIN zadanie ON pz.zadanie = zadanie.id
     GROUP BY pz.postac)
AS suma ON p.id = suma.postac;
```

id	nazwa	uzytkownik	status	klasa	miejsce	pd	pieniadze	poziom
1	Asmodeusz	_adam_	t	łotr	tawerna	780	500	6
2	Gajusz	janek23	t	druid	las	200	100	1
3	Hypatia	kasia	f	druid	ruiny	800	500	6
4	Stefan	OleK	t	bard	miasto	280	200	2
5	Lucek	kiko	t	łowca	miasto	0	0	0