

PROJEKT RELACYJNEJ BAZY DANYCH

Gra MMORPG

E1. . Pisemne opracowanie punktów A i B cyklu projektowego

A. Określenie przeznaczenia zaprojektowanej relacyjnej bazy danych (RBD)

Baza danych mojego projektu ma za zadanie służyć grze MMORPG w świecie fantasy. Dzięki jej wykorzystaniu będziemy mogli przechowywać wiele niezbędnych informacji dla gry tego typu. Przede wszystkim o graczach, ale także o świecie gry oraz relacjach między poszczególnymi elementami. Postaci graczy mają dostęp do przedmiotów, mogą posiadać swój własny kąt, mogą przynależeć do gildii zrzeszających wielu graczy. Każdy gracz może wybrać swoją przynależność do konkretnej rasy, płci. Może wybrać czy chce być wojownikiem czy magiem itd. Każda z tych opcji ma przełożenie na jego siłę w walce oraz mocne oraz słabe strony (statystyki, ich modyfikatory oraz umiejętności). Gracze zdobywają również punkty doświadczenia za pokonywanie przeciwników, każdy przeciwnik nagradza inną wartością doświadczenia. Moc umiejętności jest sumą kosztu many oraz kondycji.

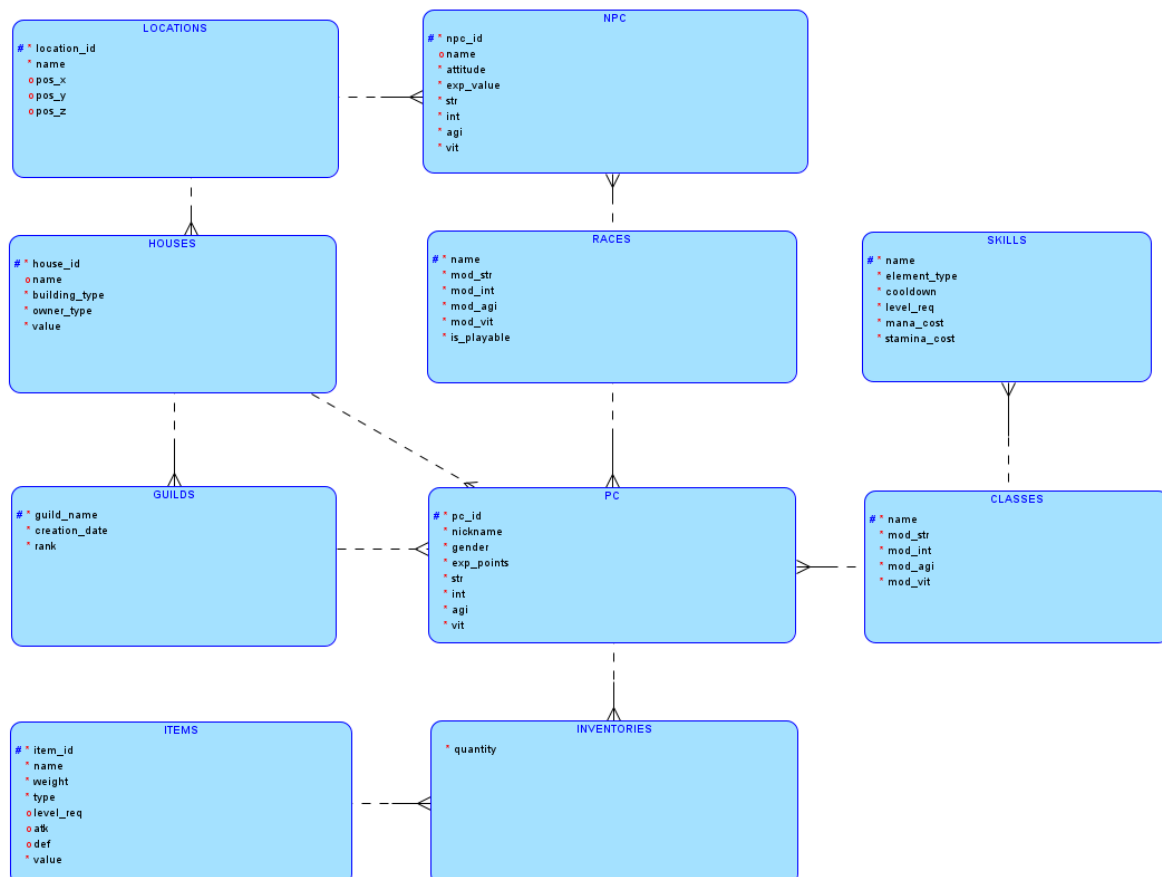
B. Przygotowanie właściwych dla wybranego tematu założeń funkcjonalnych dla zaprojektowanej RBD, które mogą zostać zastąpione sporządzeniem kompletnej listy reguł biznesowych.

- Postacie gracza PC muszą posiadać nickname, klasę postaci (class), rasę (race), punkty doświadczenia (exp_points) oraz statystyki (str - siła, int - inteligencja, agi - zwinność, vit - witalność). Posiadanie domu (house_id) oraz przynależność do gildii (guild) jest opcjonalna. Poziomy nie są jawnie prezentowane, obliczane są na podstawie posiadanych na koncie punktów doświadczenia według przelicznika 1000 punktów -> 1 poziom.
- Postacie niegrywalne NPC muszą mieć identyfikator (npc_id) oraz statystyki podobnie jak postać gracza, zakładamy że postać npc może być kimś z kim można wejść w dialog albo naszym przeciwnikiem – parametr wymagany (attitude), postać npc musi mieć wartość punktów doświadczenia (exp_value), które są dodawane graczowi za zwycięstwo, npc musi znajdować się w konkretnej lokacji (location_id) oraz wymagana jest jego rasa (race).
- Rasy muszą mieć nazwę (name), modyfikator do siły, inteligencji, zręczności, witalności (str, int, agi, vit), muszą mieć także znacznik na to czy jest to rasa grywalna dla postaci gracza – parametr (is_playable).
- Klasy postaci muszą mieć nazwę oraz poszczególne modyfikatory do statystyk (str, int, agi, vit).
- Umiejętności muszą posiadać nazwę (name), żywioł (element_type), czas odnowienia (cooldown), koszt many (mana_cost), koszt kondycji (stamina_cost), wymagany poziom (level_req) oraz klasę postaci z jaką są związane (class). Ilość posiadanej many oraz kondycji i szybkość ich odnowy zależy od statystyk gracza. Na przykład ilość many to wartość aktualnej inteligencji (int), a szybkość odnowy to ilość many / (10 * lvl) albo jakiś podobny wzór, który pozwoli uzależnić te parametry od siebie. Nie jest to na sztywno zaimplementowane do którejkolwiek z tabel, ale może zostać dodane jako widok, podobnie do poziomu, który jest obliczany na podstawie punktów doświadczenia.

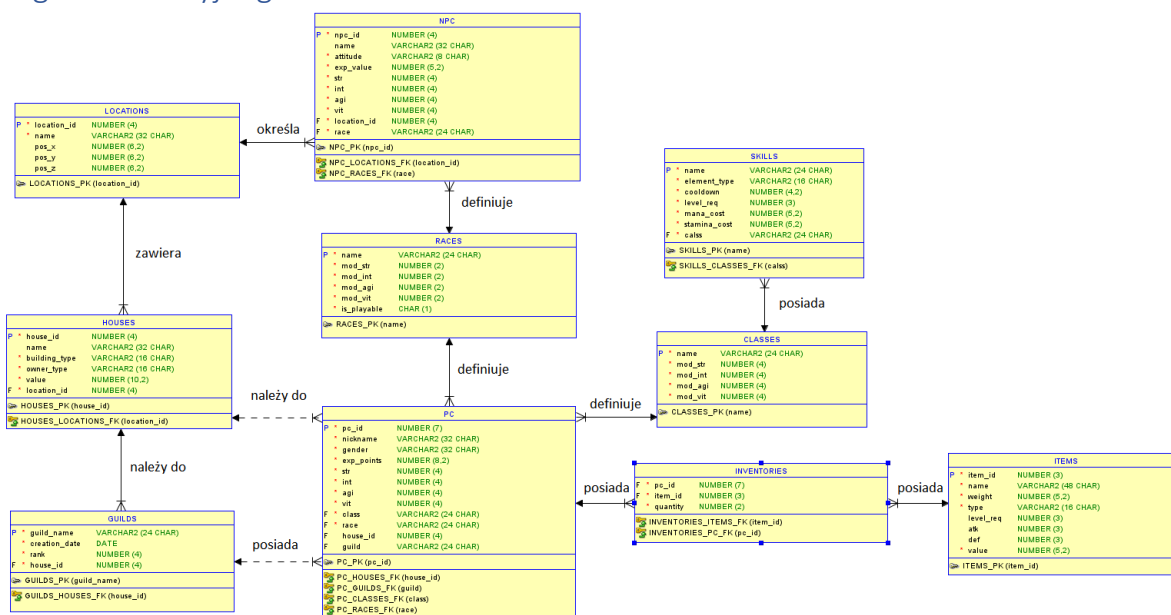
- Ekwipunki postaci są przechowywane w tabeli INVENTORIES, musi zawierać id gracza (pc_id), identyfikator przedmiotu (item_id) oraz jego ilość (quantity).
- Przedmioty muszą mieć id (item_id), nazwę (name), wagę (weight), typ przedmiotu (type) oraz wartość (value). Mogą mieć statystykę ataku (atk) oraz obrony (def) i wymagany poziom (level_req).
- Gildie muszą mieć nazwę (guild_name) która służy do identyfikacji, datę utworzenia (creation_date) oraz posiadać rangę (rank) im wyższa wartość tym niżej w rankingu, ranga jest pewnym rodzajem klasyfikacji gildii ze względu na siłę. Gildie muszą również posiadać dom, w którym gracze mogą się naradzać, wchodzić w interakcje ze sobą itd.
- Domy muszą posiadać identyfikator, typ budynku (building_type), oraz typ właściciela (owner_type) – może być to gracz lub gildia, jest to pozycja, która określa możliwość wejścia w posiadanie przez konkretną jednostkę – gracza lub gildię. Musi posiadać również wartość (value) oraz lokalizację w której jest umiejscowiony (location_id). Nazwa jest opcjonalna.
- Lokacje muszą posiadać identyfikator (location_id) oraz nazwę (name). Dopuszczalny jest brak podania pozycji na mapie (pos_x, pos_y, pos_z), ponieważ mogą znajdować się poza standardową mapą lub pozycja jest nie do określenia.

E2. Diagramy wykonane w Oracle SQL Developer w ramach punktów C1 i C2.

C1. Sporządzenie za pomocą Oracle SQL Developer logicznego diagramu RBD, który ma być przygotowany w formie diagramu związków encji (ERD).



C2. Sporządzenie za pomocą Oracle SQL Developer na podstawie diagramu logicznego diagramu relacyjnego RBD.



E3. Diagramy wykonane w ramach punktu C3

C3. Sporządzenie diagramów odpowiednich zależności funkcyjnych dla każdej relacji zaprojektowanej RBD, które wykażą wymagany stopień ich normalizacji.

E4. Skrypty przygotowane w ramach realizacji punktów D1, D3 i D5

D1. Wygenerowanie za pomocą Oracle SQL Developer skryptów z komendami DDL SQL, zawierającymi definicje obiektów zaprojektowanej RBD.

Załącznik nr. 1. Plik DDL

D3. Przygotowanie skryptu wypełniającego przykładowymi danymi utworzony schemat bazy danych, który m.in. pozwoli sprawdzić poprawność narzuconych na tabele tej bazy danych więzów integralności. W skrypcie należy uwzględnić każdą zaprojektowaną tabelę.

Załącznik nr. 2. Skrypt tworzący rekordy

D5. Przygotowanie w postaci skryptu przynajmniej 5 zapytań SQL, w których zostaną rozwiązane ciekawe i zaawansowane zagadnienia eksploracji danych składowanych w zaprojektowanej RBD. Zapytania wymagane w tym punkcie mogą zostać przedstawione jako zapytania definiujące perspektywy.

Załącznik nr. 3. Zapytania do bazy danych

Wszystkie załączniki załączam jako osobne pliki.