PROIECT BAZE DE DATE - TOMESCU LIVIU

Tema: Baza de date pentru un caractere RPG

Student: Tomescu Liviu

Grupa: 211

**1.Descrierea modelului real:**

Un developer independent lucreaza la un joc de tip RPG si are nevoie de o baza de date. Aceasta va fi folosita pentru stocarea datelor despre iteme, modul lor de obtinere, mobi\*, NPC-uri\*,misiuni precum si inventarul caracterelor pe care le controleaza jucatorul.

mob\* - inamic pe care il intalnesti in lumea jocului;

NPC\* - non player character;

drop\* - recompensa pe care o primesti la terminarea unui quest sau infrangerea unui mob

**2. Constrangeri**

**maincharacter:** main\_id, level, max\_hp, max\_energy nu pot sa fie NULL;

**inventar:** inventar\_id si main\_id nu pot sa fie NULL;

**pet:** pet\_id nu poate sa fie NULL;

**inventar\_continut:** inventarcont\_id, inventar\_id si item\_id nu pot sa fie NULL;

**item**: item\_id nu poate sa fie NULL;

**mob\_drop:** mobdrop\_id, item\_id si mob\_id nu pot sa fie NULL;

**mob:** mob\_id si max\_hp nu pot sa fie NULL;

**dungeon\*:** dungeon\_id nu poate sa fie NULL, mapzonex si mapzoney nu pot sa coincida cu alt entry din tabelul dungeon sau tabelul oras;

**quest\_drop:** questdrop\_id, item\_id si quest\_id nu pot sa fie NULL;

**quest:** quest\_id nu poate sa fie NULL;

**npc:** npc\_id si oras\_id nu pot sa fie NULL;

**oras:** oras\_id nu poate sa fie NULL, mapzonex si mapzoney nu pot sa coincida cu alt entry din tabelul dungeon sau tabelul oras;

**3/5. Descrierea entitatilor si a atributelor acestora**

**item:**

Obiecte care fac caracterul sa fie mai puternic.

atribute:

* item\_id(PK) - identificator (int);
* nume: Numele implicit al obiectului (char());
* tip: poate avea valoare de “arma” sau “armura”, in functie de valoare schimba sensul valorii stat\_value (char());
* stat\_value: in functie de atributul tip poate insemna daune bonus in cazul tip = arma sau rezistenta bonus in cazul tip = armura (int);
* gold\_value: valoare monetara a obiectului. (int);

**mob\_drop:**

Este tabelul pentru recompensele posibile la invingerea unui mob.

* mobdrop\_id(PK) (int);
* item\_id (FK) - stabileste relatia cu entitatea “item” (int);
* mob\_id(FK) - stabileste relatia cu entitatea “mob” (int);
* gold: recompensa de gold la invingerea unui mob (int);

**mob:**

Tabel pentru retinerea datelor despre mobi.

* mob\_id(PK) - identificator (int);
* dungeon\_id(FK) - stabileste relatia cu entitatea “dungeon”, poate sa fie NULL in cazul in care un mob nu apartine unui anume “dungeon” (int);
* level - nivelul (int)
* xp - valoare de xp care o primeste caracterul jucatorului atunci cand invinge mobul (int);
* max\_hp - Puncte de viata (int);
* dmg - daunele pe care le cauzeaza cu fiecare atac (int);

**dungeon:**

Tabelul pentru retinerea pozitiei, numelui si tipului de dungeonuri. Un dungeon este locul unde intalnesti deobicei mobi.

* dungeon\_id(PK) - identificator (int);
* nume: Numele dungeonului (char());
* mapzonex/mapzoney : Coordonatele intr-un spatiu 2D ale dungeonului. De precizat ca nu se pot afla 2 dungeonuri sau un oras si un dungeon la acelasi set de coordonate (int);
* biome: tip de dungeon; ex: mlastina, padure, campie; (char)

**quest\_drop:**

Acelasi tip de tabel ca si “mob\_drop” numai ca pentru questuri.

**quest:**

Tabel pentru retinerea datelor despre questuri.

* quest\_id(PK) - identificator (int);
* nume - Numele questului (char());
* started - Daca un quest este in curs de desfasurare are valoare 1, in caz contrar are valoare 0 si in cazul in care a fost facut deja are valoare 2 (int);
* startdate: Data la care a fost inceput questul. NULL daca nu este inceput (date);
* enddate: Data la care a fost terminat questul. NULL daca nu este terminat (date);
* npc\_id(FK): stabileste relatia cu “npc”

**npc:**

Tabel pentru retinurea datelor despre npcuri.

* npc\_id(PK) - identificator (int);
* oras\_id(FK) - stabileste relatia cu entitatea “oras” (int);
* nume: Numele NPC-ului(char());
* quest\_finished: 1 daca questul NPC-ului este terminat sau 0 in caz contrar(int)

**oras:**

Oras este tabelul pentru retinerea pozitiei si numelui oraselor. Un oras este locul unde intalnesti deobicei NPC-uri.

* oras\_id(PK) - identificator (int);
* nume: Numele orasului (char());
* mapzonex/mapzoney : Coordonatele intr-un spatiu 2D ale dungeonului. De precizat ca nu se pot afla 2 orase sau un oras si un dungeon la acelasi set de coordonate (int);

**inventar\_continut:**

Tabel in care se inregistreaza practic toate inventarele caracterelor.

* invcont\_id(PK) - identificator (int);
* inventar\_id(FK) - stabileste relatia cu “inventar” (int);
* item\_id(FK) - stabileste relatia cu “item” (int);

**inventar**

Tabel pentru retinerea datelor despre inventarul unui caracter

* inventar\_id - identificator (int)
* main\_id - stabileste relatia cu “maincharacter”
* max\_capacity - numarul de iteme pe care pot fii in inventar
* gold - cat gold exista in inventar (int)

**maincharacter**

Tabel pentru retinerea datelor despre un caracter

* main\_id - identificator (int)
* name - numele caracterului (int)
* level - nivelul caracterului (int)
* xp - valoare de xp care o detine caracterul (int)
* lvlupxp - de cat xp are nevoie caracterul ca sa treaca la nivelul urmator (int)
* max\_hp - puncte de viata (int)
* max\_energy - puncte de energie (int)
* gender - genul personajului: 1 - barbat, 2 - femeie(int)
* creation\_date - data la care a fost creat caracterul (datetime)

**pet:**

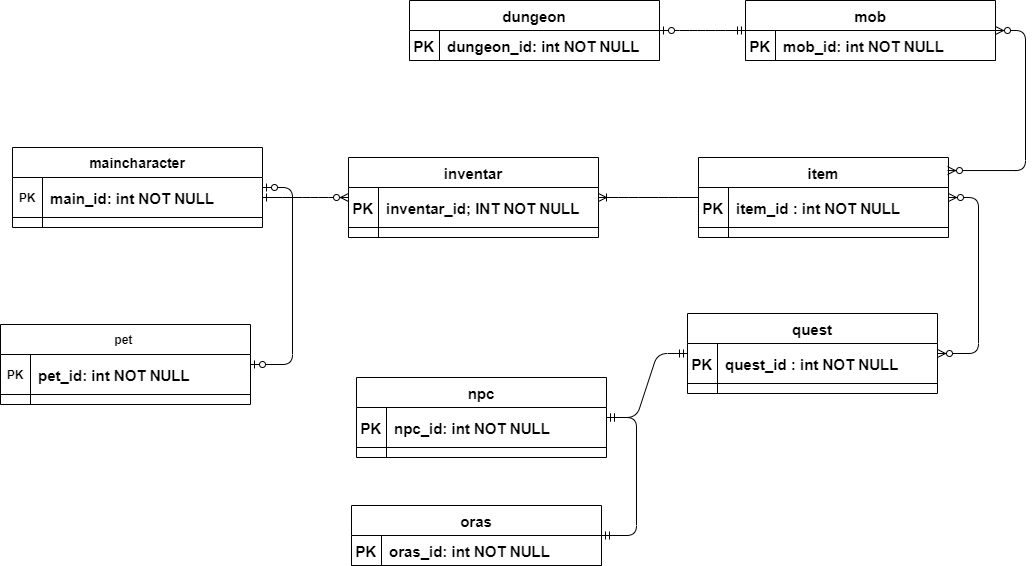
Tabel pentru inregistrarea informatilor despre animale de companie, orice aventurier trebuie sa aiba un sidekick.

* pet\_id(PK) - identificator (int);
* main\_id(FK) - stabileste relatia cu “maincaracter”. De notat, poate sa aiba si valoare 0 in cazul in care pet-ul nu este luat de niciun caracter (int);
* max\_hp - Puncte de viata (int);
* gender - genul animalului de companie (int);
* name - Numele (char());
* acquire\_date - data la care un animal de companie este adoptat de catre un caracter (date);

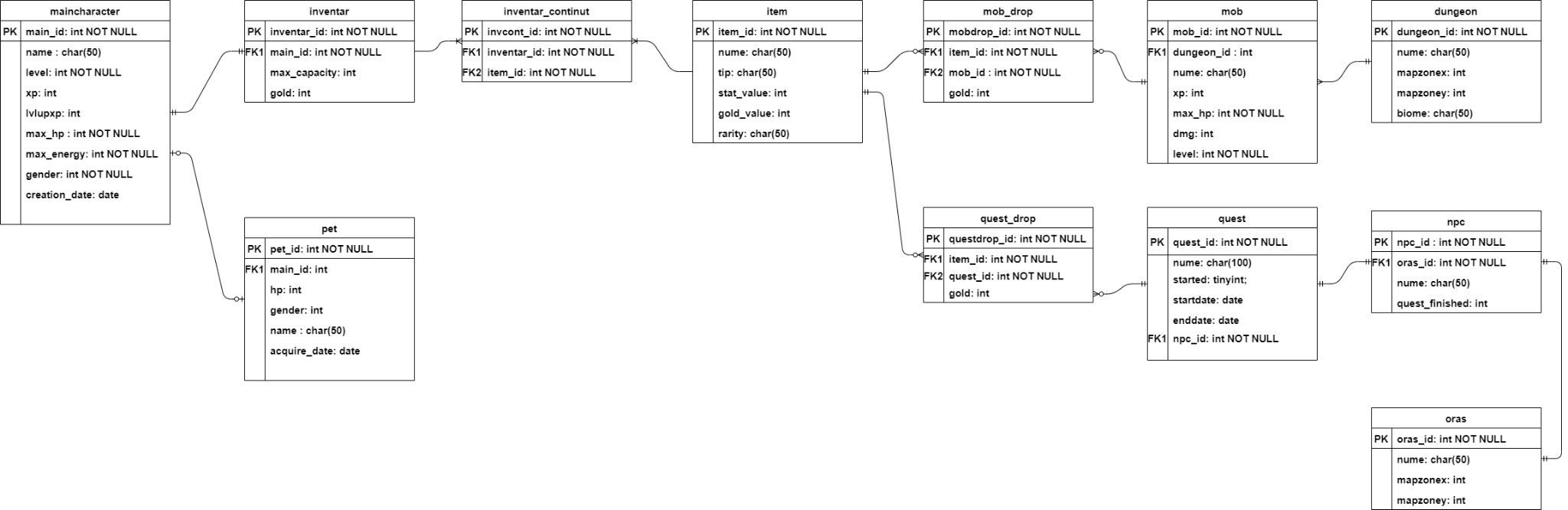
**4. Descrierea relatiilor**

* maincharacter - inventar: relatie 1 to 1. Fiecare caracter are cate un inventar unic
* maincharacter - pet: relatie optional 1 to optional 1. Nu este oblicatoriu ca un pet sa aiba un caracter sau de un caracter sa aiba un pet.
* inventar - inventar\_continut: relatie 1 to many. inventar\_continut retine fiecare item pe cate un row iar un inventar poate avea mai multe iteme.
* inventar\_continut - item: relatie many to 1. inventar continut poate sa aiba mai multe rowuri cu acelasi item\_id, deoarece un item se poate regasi in mai multe inventare
* item - mob\_drop: relatie 1 to optional many. In jocuri de tipul RPG, la infrangerea unui mob acesta are sansa de ati da un drop
* mob\_drop - mob: relatie optional many to 1. Un mob poate avea mai multe dropuri sau niciunul.
* mob - dungeon: relatie many to 1. Pot exista mai multi mobi care se gasesc in acelasi dungeon
* item - quest\_drop: relatie 1 to optional many. Mai multe questuri pot avea ca si drop acelasi item.
* quest\_drop - quest: relatie optional many to 1. Un quest poate avea mai multe dropuri(rewarduri) sau niciunul.
* quest - npc: relatie 1 to 1. NPCurile sunt responsabile cu predarea de questuri jucatorului.
* npc - oras: relatie 1 to 1. Un NPC are asociat un oras in care il poti gasi.

**6. Diagrama entitate-relatie**



**7. Diagrama conceptuala**



**8.Enumerarea schemelor rationale:**

* maincharacter (PK: main\_id, name, level, xp, lvlupxp, max\_hp, max\_energy, gender, creation\_date)
* pet (PK: pet\_id, FK1: main\_id, max\_hp, gender, name, acquire\_date)
* inventar ( PK: inventar\_id, FK1: main\_id, max\_capacity, gold)
* inventar\_continut (PK: invcont\_id, FK1: inventar\_id, FK2: item\_id)
* item (PK: item\_id, nume, tip, stat\_value, gold\_value, rarity)
* mob\_drop (PK: mobdrop\_id, FK1: item\_id, FK2: mob\_id, gold)
* quest\_drop(PK: questdrop\_id, FK1: item\_id, FK2: quest\_id, gold)
* mob (PK: mob\_id, FK1: dungeon\_id, nume, xp, max\_hp, dmg, level)
* dungeon (PK: dungeon\_id, nume, mapzonex, mapzoney, biome)
* quest (PK: quest\_id, FK1: npc\_id, nume, started, startdate, enddate)
* npc (PK: npc\_id, FK1: oras\_id, nume, quest\_finished)
* oras (PK: oras\_id, nume, mapzonex, mapzoney)

**9. non NF:**

|  |  |  |
| --- | --- | --- |
| **dungeon** | | |
| dungeon\_id | nume | mobi |
| 1 | The spooky cave | Skeleton,  BIG Spider |
| 2 | The strange swamp | Slime,  BIG Spider |

NF1:

|  |  |  |
| --- | --- | --- |
| **dungeon** | | |
| dungeon\_id | nume | mobi |
| 1 | The spooky cave | Skeleton |
| 1 | The spooky cave | BIG Spider |
| 2 | The strange swamp | Slime |
| 2 | The strange swamp | BIG Spider |

NF2:

|  |  |
| --- | --- |
| **dungeon** | |
| dungeon\_id | nume |
| 1 | The spooky cave |
| 2 | The strange swamp |

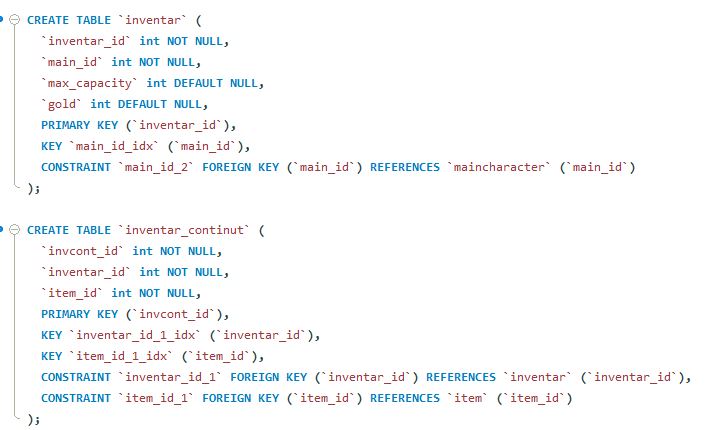
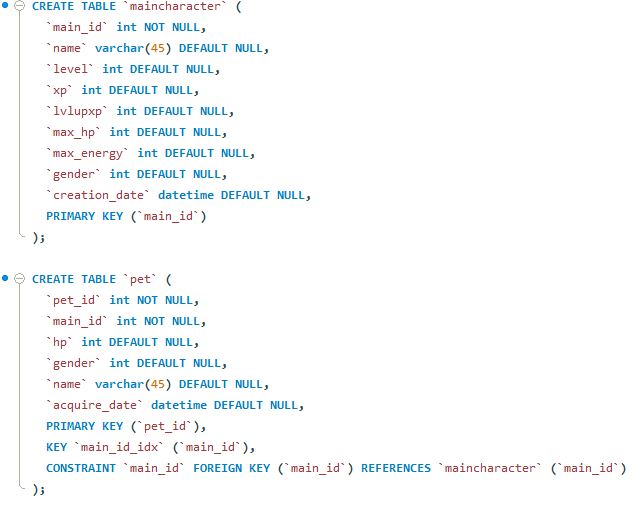
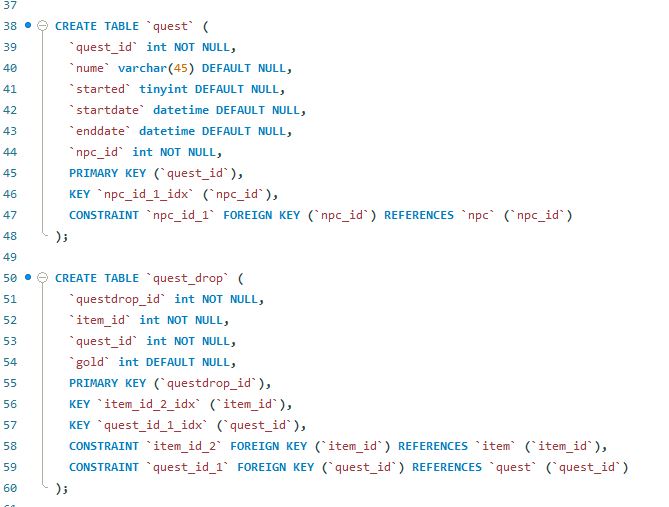
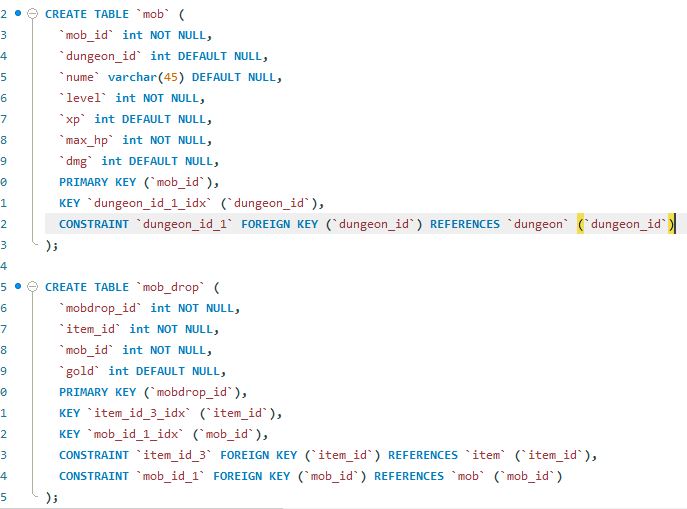
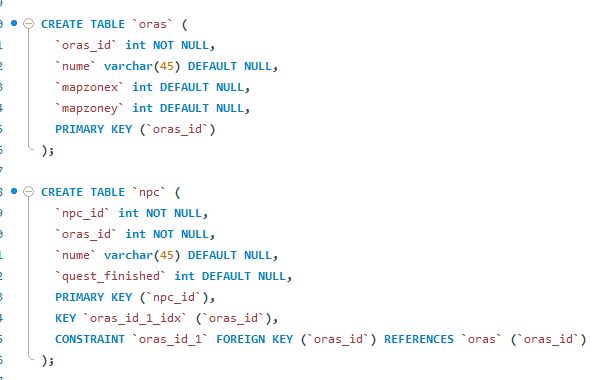
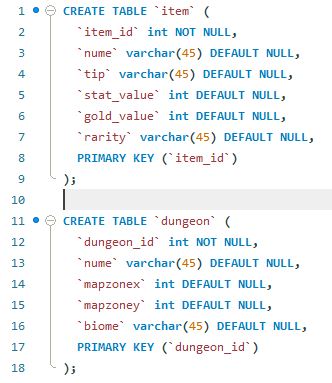
|  |  |
| --- | --- |
| **mob dungeon** | |
| dungeon\_id | mob |
| 1 | Skeleton |
| 1 | BIG Spider |
| 2 | Slime |
| 2 | BIG Spider |

NF3:

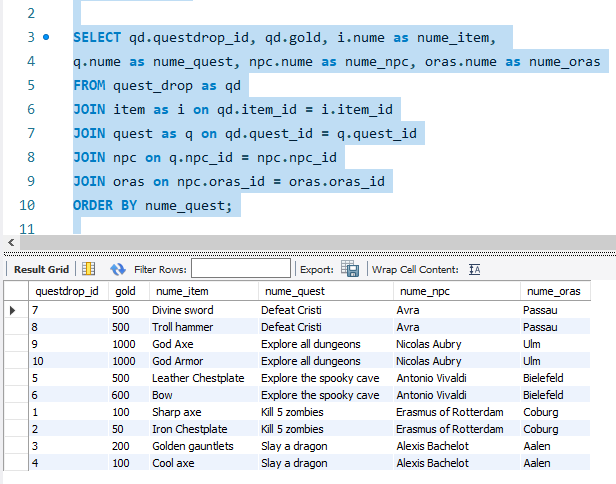
|  |  |
| --- | --- |
| **dungeon** | |
| dungeon\_id | nume |
| 1 | The spooky cave |
| 2 | The strange swamp |

|  |  |
| --- | --- |
| **mob dungeon** | |
| dungeon\_id | mob\_id |
| 1 | 1 |
| 1 | 3 |
| 2 | 2 |
| 2 | 3 |

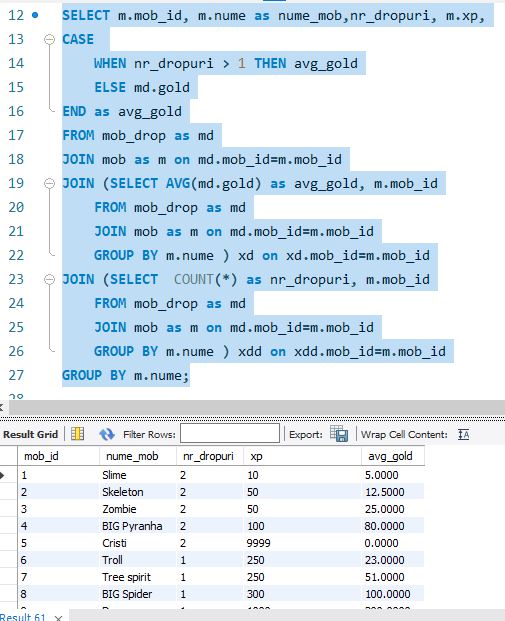
|  |  |
| --- | --- |
| **mob** | |
| mob\_id | nume |
| 1 | Skeleton |
| 2 | Slime |
| 3 | BIG Spider |

10. Crearea si inserarea tabelelor

\*Imi cer scuze pentru rezolutie, nu am facut screenshot cand le-am creat initial\*

**11. 5 Cereri SQL**

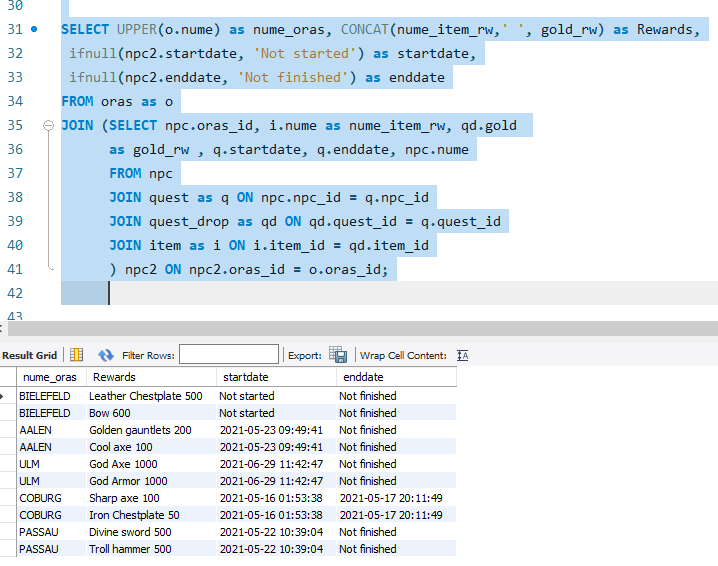
-Cerere pentru questdrop\_id, gold, nume\_item, nume\_quest, nume\_npc si nume\_oras, date luate din 5 tabele diferite. De asemenea datele sunt ordonate dupa nume\_quest. (4joinuri+, ordonare)

****

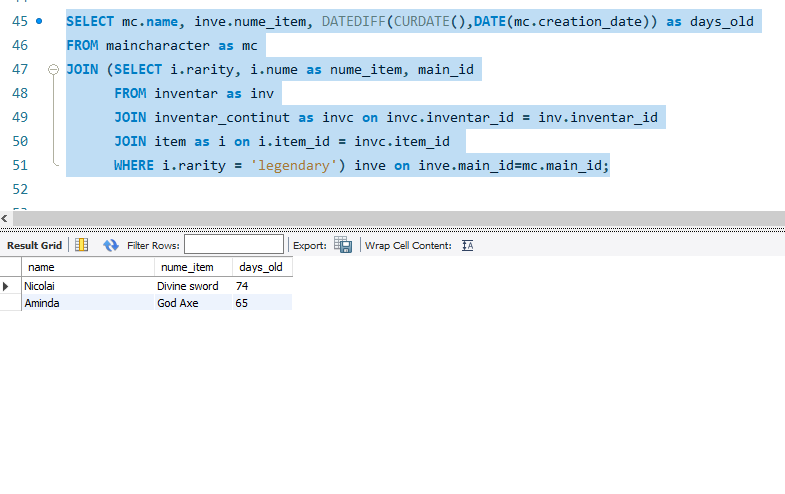
-Cerere pentru mob\_id, nume\_mob, nr\_dropuri, xp si avg\_gold.

Am folosit COUNT si AVG impreuna cu GROUP pentru a obtine valorile la nr\_dropuri si avg\_gold. De asemenea am folosit expresia CASE pentru a introduce valoare corecta in functie de nr\_dropuri in avg\_gold.

(CASE, GROUP)

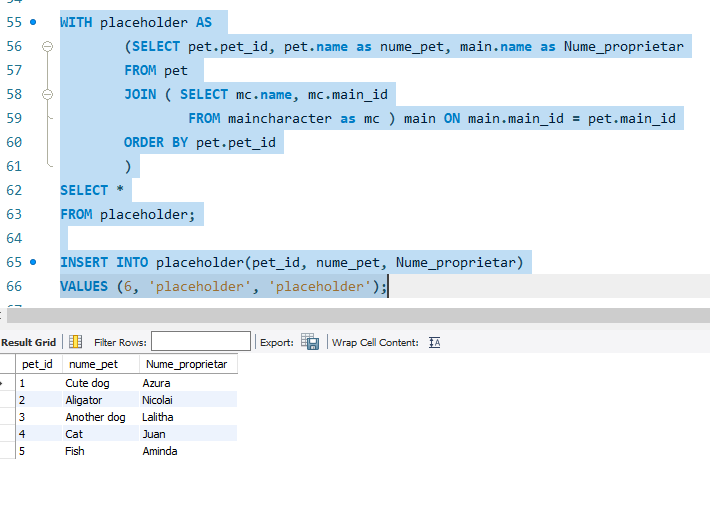
****

-Cerere pentru nume\_oras, rewards(rewardurile de la questul din oras), startdate(data cand a fost inceput questul din oras), enddate(data cand a fost terminat questul din oras). Am utilizat o subcerere cu 4 tabele pentru extragerea de date, functiile UPPER si CONCAT pentru prelucrarea datelor din nume\_oras si Rewards si funtia IFNULL(echivalentul lui NVL) pentru cazul in care un quest nu a fost inceput/terminat.

****

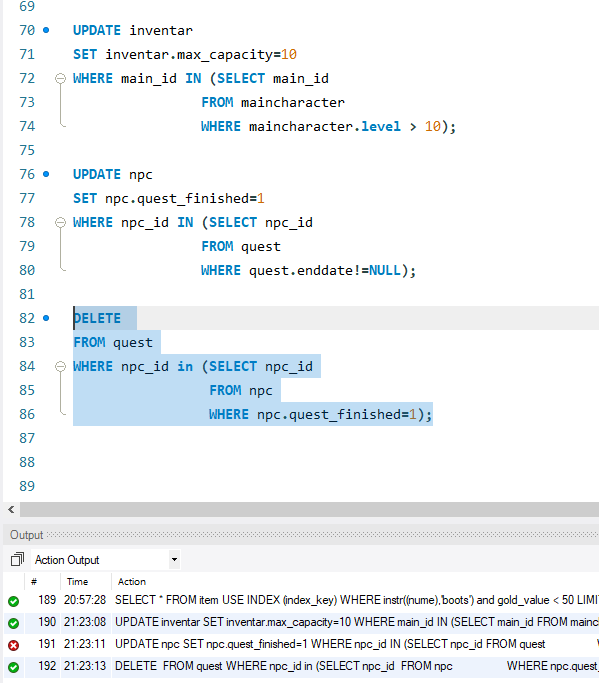
-Cerere pentru name(numele caracterului), nume\_item(numele itemului pe care il detine in inventar), si days\_old(vechimea caracterului).

Pentru obtinerea datelor despre item am folosit o subcerere sincronizata intre 3 tabele care returneaza doar numele itemelor cu rarity = “legendar”. (WHILE)

****

-Cerere care returneaza pet\_id, nume\_pet si Nume\_proprietar(Numele caracterului care detine pet-ul). Insertul este acolo pentru a evidentia ca nu putem adauga date deoarece tabelul placeholder este creat cu WITH.

**12. 3 operatii de actualizare / suprimare**

****

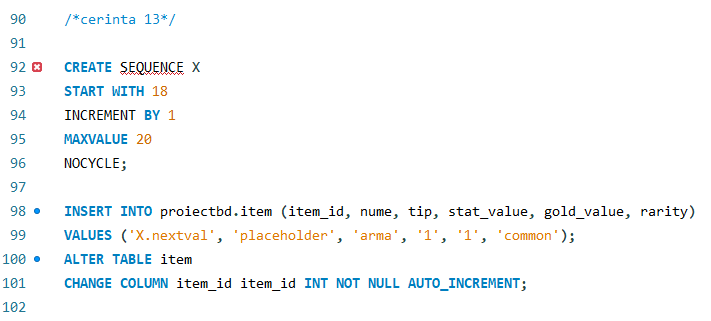
\*Al doilea update nu merge pentru ca nu stiu cum sa dezactivez safe update mode.\*

-Primul update seteaza atributul max\_capacity cu valoarea 10 atunci cand atribul level din “maincharacter” depaseste valoare 10.

-Al doilea update seteaza atributul quest\_finished cu valoarea 1 atunci cand un quest este terminat, adica cand atributul enddate din “quest” nu este NULL.

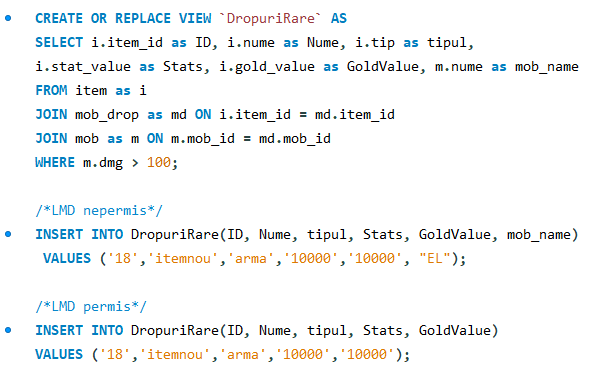
-Delete-ul sterge entry-ul din tabelul “quest” atunci cand valoarea atributul quest\_finished din “npc” are valoarea 1.

**13. Crearea unei secvente**



* Din pacate nu poti sa faci sequence in MySQL. De asemenea nu pot atribui auto-increment la item\_id deoarece este deja folosit ca si FK in alt tabel

**14. Crearea unei vizualizari compuse. LMD permisa/LMD nepermisa**



**15. Crearea unui index/optimizare pentru cautare cu 2 criterii**

