

CS 5200 Spring 2024 (Cobbe)

Term Project Data Model

For this project, you will design and build a relational database in MySQL and construct a simple web front end using Java Server Pages (JSP) and the Java Database Connectivity (JDBC) library that interacts with your database.

In this document, I describe the form of the data that you are to represent in your database. This model describes the data that a multiplayer on-line role playing game might want to keep about its players and characters.¹

Players

When a player first starts playing this game, they must create an *account*, which records their name and their email address. Once the player's account is active, they can log in and create one or more *characters*.

Characters

A character is associated with a single player account and has the following properties:

- a first name and a last name. The combination of these two properties must be unique among all characters, and the game enforces this at character creation time.
- various *attributes* that indicate the character's power in different areas
- one or more associated *jobs*; each job has a separate level and number of experience points.
- an *inventory* containing a collection of *items*
- a set of *equipped items*, corresponding to the weapon that the character is currently using and the armor or clothing they are currently wearing.
- money, in the form of several different *currencies*.

See figure 1 at the end of this document for an example of a character's properties. The left-hand side of the figure contains information about the character's attributes and their values. (In this window, attributes are divided into multiple categories, like "Attributes," "Offensive Properties," and so forth; you should not concern yourselves with these categories for this project.)

¹ This model is based on an actual MMORPG, but completing this project does not require any knowledge of the game beyond the contents of this document.

Equipped items represent the gear that a character is currently wearing. A character can, for instance, have multiple pairs of boots in their inventory, but only one can be equipped at a time. Characters can equip items in a variety of *slots*, which correspond to the parts of the body on which a particular item can be worn, so gloves go in the "hand" slot, for example, and boots on the "feet" slot.

The right-hand side of figure 1 demonstrates the various slots and equipped items; these correspond to the green icons around the character portrait. For the purposes of this assignment, you should model the following slots:

main hand	
head	off-hand
body	earring
hands	wrist
legs	ring
feet	

(As the figure suggests, the original game has a couple of additional slots, but we're going to ignore those for simplicity.)

As we will see below, any specific item can only be equipped in a particular slot, and a character can have at most one item equipped in any slot, for a maximum of 10 equipped items. Further, the character must *always* have something equipped in the main hand slot, but all other slots are optional. (In the figure, for example, the character does not have anything equipped in the off-hand slot.)

Characters and Jobs

Each character can play a variety of different *jobs*, such as (in the original game) Warrior, Samurai, and White Mage. The list of available jobs changes over time, as new jobs are added to the game. Characters can have a different *level* (represented as a number from 1 to 90) on each different job. The character in figure 2, for instance, has reached level 33 for the Scholar job, and level 90 on Machinist. In the screenshot, the large number listed next to each job indicates the character's level for that job.

Each character must have at least 1 level in at least one job, but beyond that, characters may or may not have earned levels in any particular job. In the screenshot, a job not yet played by the character is listed with level 0, as with Sage or Reaper in the screenshot. For instance, a newly-created character might choose Dragoon as their first job, and that character would be created with 1 level in Dragoon and no levels in any other jobs. Your data model must allow a user of your program to determine which jobs a particular character has played, although you may choose how you wish to represent this information.

The jobs are divided into different categories ("Tank," "Melee DPS," etc.); you do not need to concern yourself with these categories for this project.

Currencies

A character can earn money in a variety of currencies, as demonstrated in figure 3. There are a variety of different currencies, each with a name (a string; the names are not represented in this figure) and an amount. Additionally, each currency has a maximum amount (or *cap*) that a character may hold at any one time; the cap for a particular currency is the same for all characters. Finally, some currencies (such as the third one in the figure) also have weekly caps, corresponding to the amount of that currency that a character can earn each week. The weekly cap for a particular currency is the same for all characters.

For capped currencies, your model must track not only the total amount of the currency owned by a character (230 of the capped currency in figure 3), but also the amount of the currency earned by the character during the current week (20, in figure 3).

Some currencies have been discontinued, as in the diagram; you do not need to represent these for your project.

The set of available currencies changes over time, as new currencies are added and old ones are discontinued. You can see an example of a discontinued currency in the screenshot; your model does not need to include information about the amount a character owns of discontinued currencies.

Inventories

Each character has an inventory, or collection of items. An inventory (as displayed in figure 4) consists of a number of slots, in which characters can store *stacks* of items. Each stack is a collection of multiple "copies" of the same item, and the inventory tracks the number of items in a particular stack, as indicated in the figure. (A missing number indicates that the slot contains a single item, or, equivalently, a stack of 1 item.) Note that in a stack, all of the items must be "the same:" you can have a stack of 62 raisins, as highlighted, but you cannot have a stack that contains 62 raisins and 5 apkallu eggs—different kinds of items must be stored in different stacks.

Among the various properties that items have (discussed further below), all items have a *maximum stack size*: this indicates the maximum number of "copies" of an item that can be stored in a single stack. As an example, most gear has a maximum stack size of 1 (meaning that it effectively cannot be stacked); many consumables stack to 99 or even 999.

The position of the item in the inventory—i.e., which slot it occupies—is also important, and you must represent it in the database. For the purposes of this project, you can treat the inventory as a 1-dimensional sequence of slots, numbered 1, 2, ...

Items

The game includes a variety of different items that characters can hold, equip, and use. All items have certain common properties:

- item name
- the item's maximum stack size (as discussed above)
- a vendor price: the amount of gold that a character receives when selling the item to a vendor.² Note that not all items can be sold to a vendor, and your database must be able to represent this fact.

Items fall into several categories:

- gear
- weapons
- consumables

Each of these has its own set of properties, in addition to the common properties described above.

Items: Gear

Gear refers to all items that can be equipped by a character, with the exception of weapons. In addition to the common item properties above, gear items have the following properties:

- item level (a rough indication of the item's power level)
- which slot the item is equipped in
- a list of one or more jobs that can equip the item
- a required level (a character must have reached at least this level to use the item)
- some number of attribute bonuses, such as "Strength +354."
- a defense rating
- a magic defense rating

See figure 5 below for an example item, although do note that this screenshot includes a great many properties that you should *not* represent in your database.

² Gold is, technically speaking, a specific currency, but since vendors only ever give gold for items, you don't need to represent the identity of the currency as part of the vendor price.

Items: Weapons

Weapons are, strictly speaking, a type of gear, but they have several properties that apply only to them. Further, weapons can only be equipped in the main-hand slot, and *only* weapons can be equipped in the main-hand slot. In addition to the general properties that apply to all items, weapons have the following properties:

- item level
- a list of one or more jobs that can equip the weapon
- a required level
- some number of attribute bonuses, such as "Vitality +58."
- damage done
- auto-attack
- attack delay

See figure 6 below for an example weapon and its properties.

Items: Consumables

Consumable items are things like food or potions that characters can consume for temporary effects. In addition to the various properties shared by all items, consumables have the following properties:

- item level
- description
- attribute bonuses

Consumable attribute bonuses are not quite the same as gear or weapon bonuses: these augment a character's ability by a percentage of the character's base value for that ability, but with a cap. To take the example item in figure 7, consuming giant popoto pancakes raises the character's Tenacity attribute by 8%, but by no more than 61. (So, for example, if a character has a Tenacity attribute of 500, consuming this food would raise the attribute by 8% of 500 = 40, but if the character's Tenacity attribute is 800, consuming the food would only raise the attribute by 61, since 8% of 800 is 64, higher than the cap.)

Also, note that *all* consumables have an EXP bonus of +3% and an effective time of 30 minutes, so you do not need to store these properties.

Figures



Figure 1: Character Attributes and Gear



Figure 2: Jobs

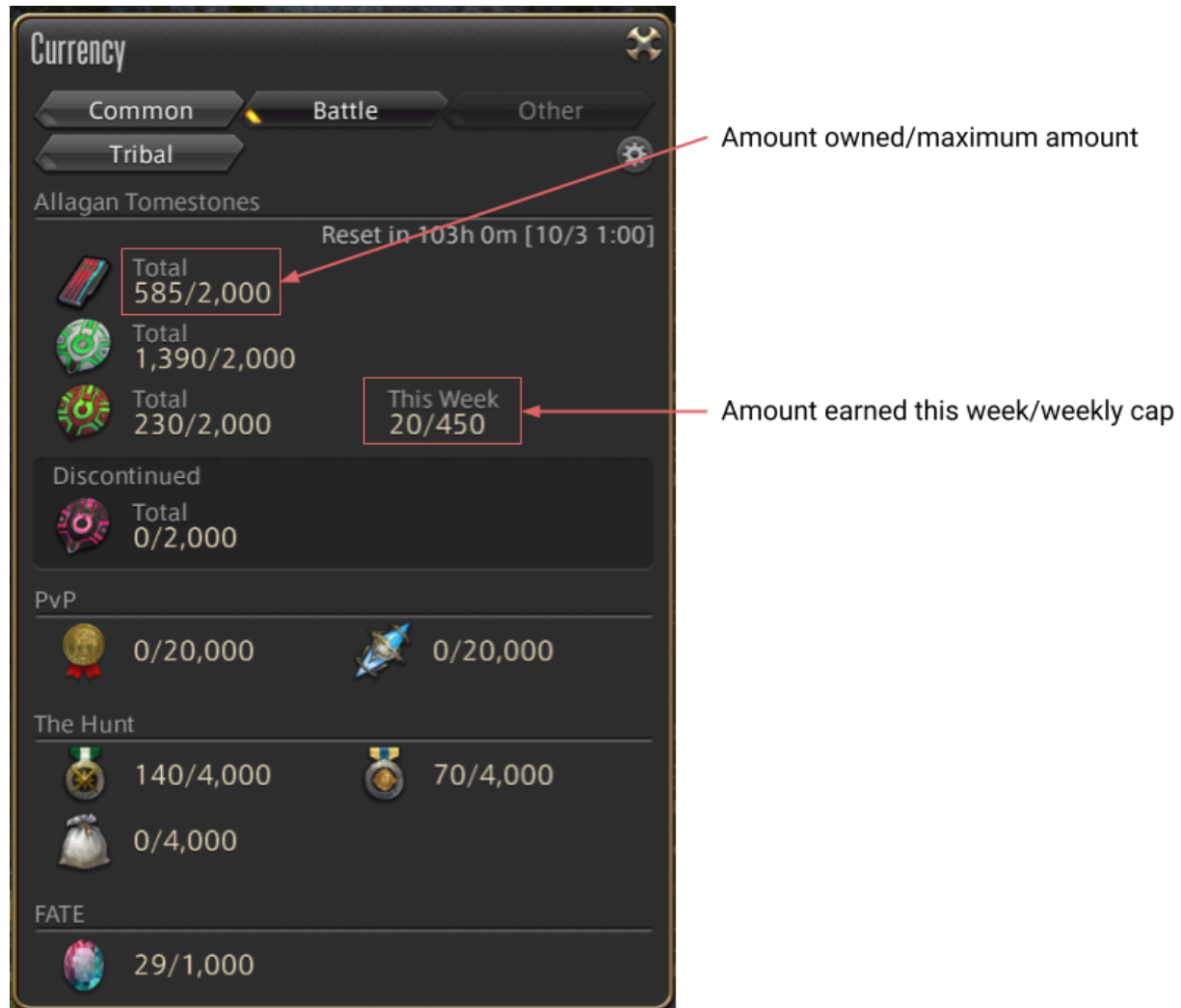


Figure 3: Currencies

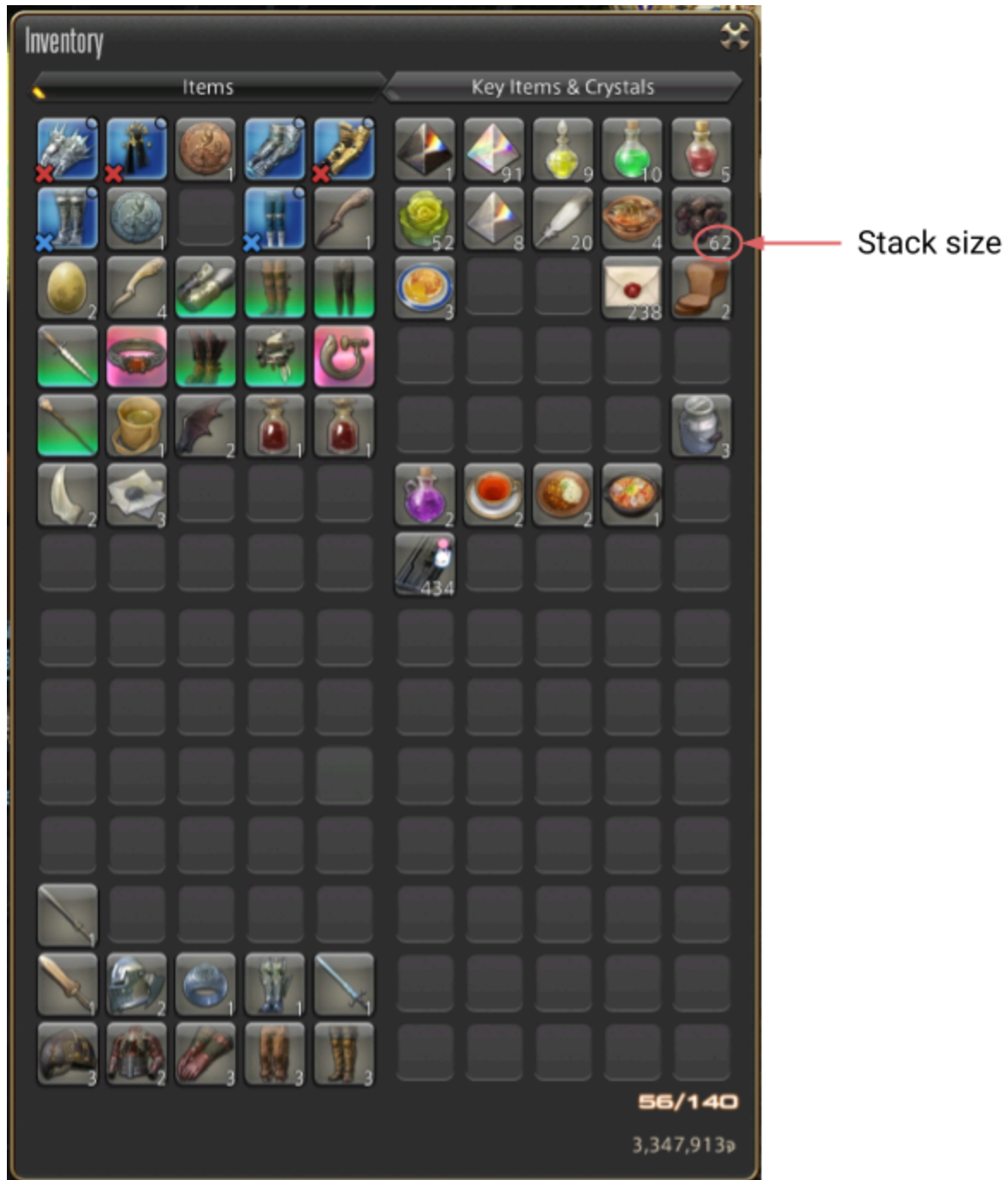


Figure 4: A sample inventory

BINDING

Name → **Diadochos Jacket of Fending**

Gear slot → **Body** (Total: 0 / 1)

Defense **1146** Magic Defense **1146**

Allowed jobs → **ITEM LEVEL 640**
→ **GLA MRD PLD WAR DRK GNB**

Required level → **Lv. 90**

Soot Black Dye

Bonuses

Strength +354	Vitality +387
Critical Hit +196	Determination +280

Materia

Savage Aim Materia X	Critical Hit +36
Savage Aim Materia X	Critical Hit +36

0 Crafting & Repairs

Condition 89%
Spiritbond 6%
Repair Level Leatherworker Lv. 80
Materials Grade 8 Dark Matter
Quick Repairs 71p
Materia Melding Disciple of the Hand Lv. 90
Extractable: Y Projectable: Y Desynthesizable: 640.00
Shop Selling Price: None

Vendor price → **Sells for 858 gil** Market Prohibited

Figure 5: An example gear item

UNIQUE UNTRADABLE

Augmented Ironworks Magitek Spear

Lancer's Arm (Total: 1)

Physical Damage	Auto-attack	Delay
65	64.13	2.96

ITEM LEVEL 130

LNC DRG

Lv. 50

Bonuses

Strength +54	Vitality +58
Determination +36	Direct Hit Rate +51

0 Crafting & Repairs

Condition 93%

Spiritbond 0%

Repair Level Blacksmith Lv. 40

Materials Grade 5 Dark Matter

Quick Repairs 13p

Extractable: Y Projectable: Y Desynthesizable: 130.00

Shop Selling Price: None

Sells for 695 gil Market Prohibited

Weapon properties

Figure 6: Weapon properties

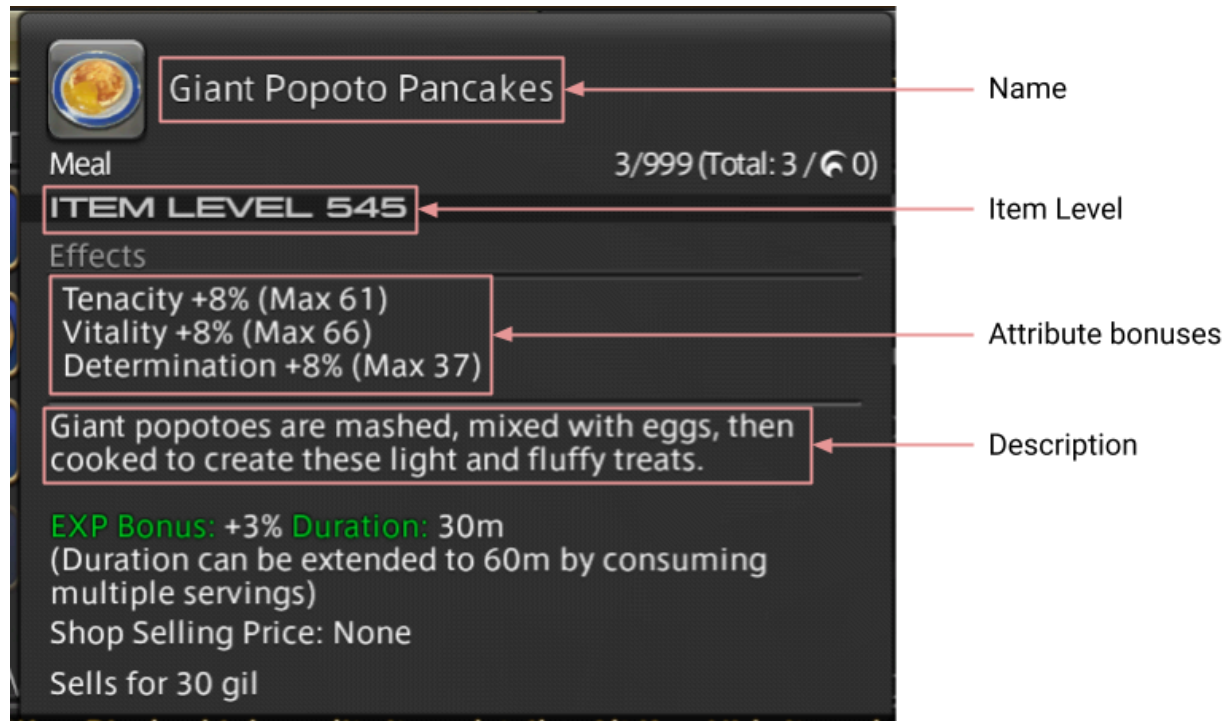


Figure 7: A consumable item