Dean Branaman, Andrew Spalla

CS 340

19 November 2019

URL LINK: flip2.engr.oregonstate.edu:4556

Copper Condors - Step 5 Draft

Feedback by Peer Review - Part 4

Jon Paulo Bautista

You've seemed to get the sql part of the assignment down. I didn't find any problems.

Regarding the website: I guess it technically follows the guidelines, but it looks like the only way to view the full list of every table will be through OSU's actual database website rather than the website you've made.

The more I look at this, the more I realize that you've included everything, but I found that it's very hard to track how everything is implemented since it seems that your UI is made to view things from a player's perspective and not an admin perspective. Combined with the fact that there's a "Dev" section, it makes things a bit confusing.

Without being able to see the full list of data on the website, I feel like it'd be a very tedious process to go back and forth between the phpMyAdmin page and your personal website to verify if all of the CRUD implementations work correctly. Even though the project guidelines don't specifically state this scenario, I think that this method would also prevent the graders from verifying that your CRUD functionalities work, so I think that it's something to think about.

Richard Miller

Hi Dean and Andrew,

Your DDQ file loaded with no issue. At a glance the DML looks correct.

Raymond Lieu

Hi Andrew and Dean,

The definitions were imported without issues. I'm running into issues with the manipulations though. For example, for delete character, there is a Player.name, but I don't see a name attribute in your Player entity. I tried changing it to Player.user_name but I couldn't run it either.

Also had issues running the Insert into Character one.

Maybe also add unique constraints to name if you're aiming for that.

Actions based on feedback

Jon's comments highlighted an important aspect of usability in our website that accessed our database. Although we had followed guidelines for the project, our implementation restricted users from effectively being able to correctly verify our database queries and manipulations. As such, we implemented data to the user at the cost of "realism" for the added benefit of the user.

Raymond's comments highlighted several typos presented in the manipulation SQL file. These typographical errors were corrected.

Upgrades to draft version

The ERD and discussion paragraphs regarding the various tables in our database were updated to correctly match out database schema.

Updates from creating our SQL files

In creating our DML and DDL queries we found a few discrepancies in our ERD and schema that slipped through part 3 and updated them accordingly. We also changed the race and class attributes of the character table from integers to varchar to better fit the selection method on our website and removed the number of raids beaten character from the guild table.

Feedback by Peer Review - Part 3

Justin Phillips

The site is off to a great start it is easy to navigate but not to sure what the simulator part is about. I just started wow classic so I am a big fan of where you are

going with this. I think a nice feature would be to have a table of all the guilds and raids available also maybe a list of your characters that are already created.

Justin Parks

Hi there! I really like the idea for this database. I've always been a fan of MMO's, and this feels like the start of the raid group sites I used back in the day. I like that filtering is the core focus of the site, but I would consider putting some filler data above or below the filtering sections to give users an idea of how their filtering will affect the info they see. Additionally, I don't see an area to delete entries in a database, but you might just be adding that later. Overall, I like where the site is going!

Alexander

This already looks very interesting. I see you can create and delete a character, create a guild, create a raid, and create a player (user account). Deleting a character would make sense if you were already logged in, but enforcing that might be too complicated for this project. When you create a new raid group, does the user automatically get assigned to being the leader of the new group, or will it be empty? This is an important design decision. Also, make sure that when the guild and raid group forms are updated, the characters are also updated to reflect if they are still a member of the guild or raid group. If the user table was not updated as well as the guild/raid group tables, the database would inaccurately reflect the users' choices.

Zhiyi He

Your website gave us a good guide to the UI, making me think on whether our UI is too focused on displaying what is already in the database and forgetting to provide the function for users to input data (even if it's just a draft). You can try to add the functions that display and selection of database content, which will make your website more perfect.

Herbert Diaz

Hey Dean and Andrew! Just giving some extra feedback after my brief overview - be aware that there's always a chance I may have missed something you have actually implemented.

- Regarding SELECTs, I assume a lot of these forms will use SELECT queries to validate user entries. I couldn't find a way to view some entries, like how to view info from the "Guild_RaidGroup" table.
- Regarding INSERTs, I couldn't find a way to create a new raid.

- Regarding DELETEs, I couldn't find a way to DELETE from an M:M relationship.
- Regarding UPDATEs, I couldn't find a way to edit any entries.
- I saw that entering a Raid Leader could be NULL, but it seems that server side would select someone preventing it from being NULL. As such, I couldn't find a NULLable relationship.

Sounds like a cool LFG web application.

Actions based on feedback

Several of the comments we received pertained to the visual aspect of either outputting a sample template for data or in fact providing some sample data to help visualize the potential outputs from the database interaction. We deemed these as valid and agree that it would better showcase out website, however for the purposes of the assignment requirements at this stage, we have opted to disclude them at this time. One student comment as well as an instructor comment highlighted the fact that our current website layout has no ability to manipulate or show data pertaining to raid groups. This was originally left out for the sake of providing a realistic database. However, based on project requirements, we have elected to add a “dummy” administrative page that would otherwise be hidden to a typical user to allow for database manipulation of this particular table. While we originally planned on the raid leader to fulfill our NULLable requirement Herbert Diaz correctly observed that it is not truly a NULLable field. Reviewing our schema a character can be created without a guild, this would actually be the norm, and as such fulfills the NULLable criteria even though it is not explicitly stated as such.

Upgrades to the draft version

- Added an administrative web page to account for the raid group table.

Feedback by Peer Review - Part 2

Adil Shakil Chaudhry

Looks good Andrew. Visually, you may want to make the relationship lines larger in part C, they are rather hard to read even when zoomed in on the PDF. Might be easy for a TA to misinterpret the diagram if they don’t zoom in all the way to get a clear look.

Kurt Kaiser

This is great never thought of how a database could be used to model this. Why not have raid group leader also present in the raid group? The only reason I say that for all those players that are not in a raid group, they don't need to have that extra boolean stored. If you had it in the raid group as a string, you could just check if the person was the raid leader. The info you have is laid out well. Neat idea.

Herbert Diaz

Hey Dean and Andrew! Here's what I caught after a brief read-through of you draft:

- Schema
    - Your schema is missing tables that would indicate a many-to-many relationship. Look in the Week 3 Learn module under "many-to-many" for an example of how it would be represented.
- Additional Notes
    - You might want to use an ID number for some of your entities.
        - For example, a guild may change its name or there might be multiple guilds with the same name.
    - It would be helpful if you listed the specific relationships you're choosing to represent for your database in your database outline.
        - For example, the Player and Character entities have a one-to-many relationship where a player can have zero to many characters and a character can only have one player.
    - Aside from what I caught in my brief overview, be sure to double check that your entities, attributes, and relationships match your database outline.

Actions based on feedback

The point of an ERD and Schema is to convey information about the database and we agreed with the comment about bolding the relationship lines to facilitate that.

Upon further inspection we also agreed that moving the Raid Group Leader attribute to the Raid Group entity would ultimately results in less overall storage which strengthens the database. Instead of making this attribute a string type, we instead

decided to make it an integer that relates back to the Character ID. In real world cases character names are unique on a server but not unique across all servers. By referencing the Character ID we ensure the uniqueness of the raid leader.

A table to represent the Guild_Raid Group relationship was added under Herbert's suggestion. In reference to comments pertaining to the relationships being stated in the outline, we felt that their inclusion in the table under each entity description was sufficient. We did decide to highlight those fields to better show that information.

An action that we found when reviewing our outline this time was that the foreign key referencing a Character in the Player entity was not a good design as a Player may make multiple Characters. We added the Player ID attribute to the Character entity to act as a foreign key pointing towards the Player and updated the ERD and Schema accordingly

Upgrades to the draft version

ERD:

- Relationship lines bolded
- Raid Group Leader attribute moved from Character entity to Raid Group entity
- Player ID attribute added to Character

Schema:

- Attribute relation lines were bolden
- Raid Group Leader attribute moved from Character entity to Raid Group entity
- Guild_RaidGroup table added
- Player ID attribute added to Character entity and linked to Player ID

Outline:

- Relationship rows highlighted blue
- Raid Group Leader attribute removed from Character description and added to Raid Group description, type changed to integer
- Minor formatting to better align sections with page breaks
- Added Player ID attribute to Character entity and

Part A - Feedback Revisions

The feedback that our group received revolved around utilizing a data type that is not compatible with SQL: arrays. Specifically, we attempted to utilize arrays to collect the entities that were related en masse to another entity for purposes of timely data output. However, by utilizing proper relationships and search queries, a Database Management System handles this for us. Therefore we have chosen to restructure

several of our entity attributes that pertain to our previously used invalid SQL data types.

Changes:

- Player Entity
    - Added a “characters” integer attribute that will relate to a specific character entity
- Character Entity
    - Added a “raid leader” boolean attribute that when combined with a “guild name” varchar attribute, will successfully identify the raid group leader.
- Guild Entity
    - Removed the “roster” attribute that was identified as an invalid SQL data type.
    - Changed the “raids completed” attribute to an int data type from an invalid SQL data type.
- Raid Group Entity
    - Removed the “characters” attribute that was identified as an invalid SQL data type.
    - Removed the “guilds attribute that was identified as an invalid SQL data type.
- Raid Entity
    - Added a raid entity to account for different raid events in order to preserve our existing project outline while still adhering to valid SQL data types.

Part B - Updated Outline

Overview

Our database will be modeling a generic MMO (Massive Multiplayer Online game) and specifically the player and guild aspects of the game. The four entities in the database will comprise of: Players, Characters, Guilds, and Raid Groups.

Database Outline

Player Entity

The Player entity represents the person playing the game. Each player will typically have created and played multiple characters, but a brand new player may have not reach the character creation stage yet. A player creates a username and password

for their account. Each player is given a unique id number for identification purposes. They must pay for playtime in the game as marked by the last day they can play before their subscription runs out. We also track when the player's account was first created and what expansions they've bought via an account level.

Relationship: Character	Zero to Many
Attribute, key: Player ID	Int (20)
Attribute: Character	Int (20)
Attribute: User Name	VarChar (20)
Attribute: Password	VarChar (8-20)
Attribute: End of payed play time	Date
Attribute: Account creation date	Date
Attribute: Account level	Int (0-5)

Character Entity

The character is the avatar of the player in the game. A character can only be made by a player and will only ever be that one player's character. A character is eligible to join a guild but can only belong to one guild at a time. Additionally, a character may or may not be present in a Raid group, as defined by a varchar data type attribute. Characters store the players ID to link them. A character is defined by their name, class, and race primarily for game purposes. Finally, a character instance's level increases through successful raids and experience gain.

Relationship: Player	One to one
Relationship: Guild	Zero to one
Attribute, key: Character ID	Int (20)
Attribute: Guild membership	VarChar (20) - Guild name or null
Attribute: Raid group	VarChar (20) - Raid Group name or null
Attribute: Name	VarChar(15)
Attribute: Player ID	Int(20)

Attribute: Class	VarChar (20)
Attribute: Race	VarChar (20)
Attribute: Level	int(1-100)

Guild Entity

A guild is a collection of characters based around a central idea; most are formed to perform the games most difficult content, raids or dungeons, though some are formed for player versus player play, roleplaying, or just to talk to each other while playing. Although a guild requires ten characters to sign up before it is created, so long as one character remains in the guild the guild will exist (some players even pay others to create guilds just so the others can leave and leave them a guild to themselves). Guilds are not required to form raid groups but most aim to form at least one with larger guilds capable of having multiple raid groups running at the same time. Guilds are identified by a unique guild name. We also track the number of raid groups present in a guild.

Relationship: Character	One to many
Relationship: Raid group	Zero to many
Attribute, key: Guild ID	Int (20)
Attribute: Name	VarChar(20)
Attribute: Raid Groups	Int
Attribute: Number of characters	Int(1-100)

Raid Group Entity

A raid group is a group of characters formed to take on the challenging raids that the MMO games provided as endgame content. One character will form the raid group as its leader and invite others to join it. A character can only belong to one raid group at a time as they must leave to join another. Typically one guild will comprise the whole raid group but with the ability to join a random raid group for faster access to raids it is likely that a raid group will have the representation of multiple guilds from its composite characters. It is also possible to form a raid group of only guildless characters.

Relationship: Guild	Zero to many

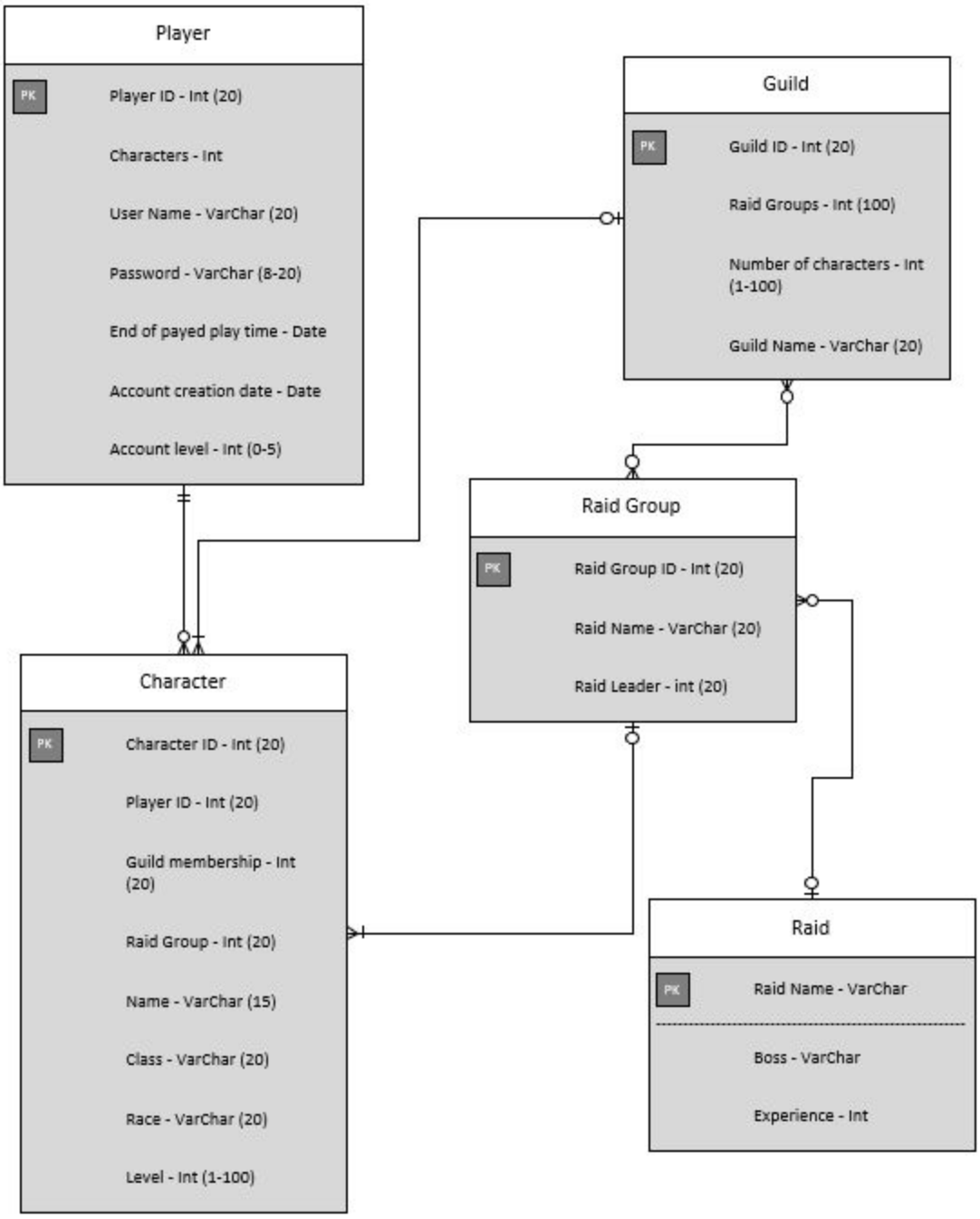
Relationship: Character	One to many
Relationship: Raid	Zero to One
Attribute, key: Raid Group ID	Int (20)
Attribute: Raid Name	VarChar (20)
Attribute: Raid group leader	Int (20)

Raid Entity

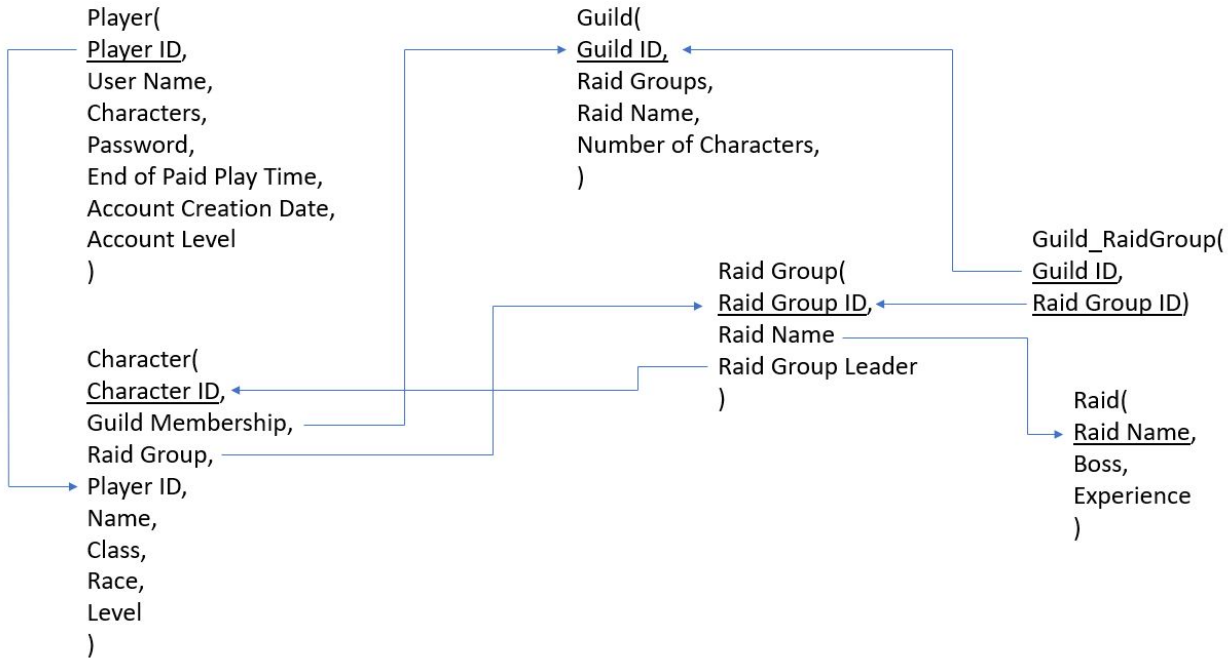
A raid is an instance of a game event that raid groups will encounter that will provide players within the raid group to gain experience.

Relationship: Raid Party	Zero to many
Attribute, key: Raid Name	VarChar
Attribute: Boss	VarChar
Attribute: Experience	Int

Part C - Entity Relationship Diagram



Part D - Database Schema


Player(
Player ID,
User Name,
Characters,
Password,
End of Paid Play Time,
Account Creation Date,
Account Level
)
Guild(
Guild ID,
Raid Groups,
Raid Name,
Number of Characters,
)
Guild_RaidGroup(
Guild ID,
Raid Group ID)
Raid Group(
Raid Group ID,
Raid Name
Raid Group Leader
)
Character(
Character ID,
Guild Membership,
Raid Group,
Player ID,
Name,
Class,
Race,
Level
)
Raid(
Raid Name,
Boss,
Experience
)