

Backend Bros Project Draft

URL:

<http://flip3.engr.oregonstate.edu:14180/>

Project Name:

Ogres and Alchemy database project

Team Members:

Richard Le
John Lemley

Overview:

Ogres and Alchemy is a (*made up*) Massive Multiple-Player Online Role-Playing Game (MMORPG) where users can role-play as a character and quest in a fantasy land with friends. Users are able create individual accounts where they choose from multiple servers where they would like to create a character to play the game. Users will be able to create characters on any server. The character will be required to select a class within the game. Also, characters will be able to obtain new weapons stored in their inventory. Ogres and Alchemy has on average 20,000 concurrent players over 5 servers and 50,000 accounts.

Summary:

There have been a lot of changes made to this project from its initial design. A more detailed look can be found at the end of this draft if you want to take a look at the specific feedback and actions taken based on the feedback. The initial version of our draft included an ERD and an outline. We didn't have the Factions entity in our design yet. Initially, we had Users, Servers, Weapons, Classes, Characters, and Inventory. This was the base we started with and from here we added more, such as the schema, creating sample data, and implementing the database in our app.

From our initial ERD, we had to make changes to the relationships to properly reflect what we wanted our entities to represent. Also we had some attribute names be changed so that it was more readable and human friendly. The early steps mainly consisted of perfecting our entity relationships so that it made sense and before we started actually building something from our diagrams.

At about the midpoint of the project, we started to build the ERD by creating the schema in MySQL. We learned how to write queries to create the tables and relationships. Also we wrote queries to include example data for the entities. We had no problems in the early stages of the project and things were progressing very smoothly.

It was when we got to implementing our database design into a web application that things became a bit more complicated. We made the mistake of trying to incorporate flask or nodeJS to create fleshed out forms, instead of simply creating the UI and leaving the implementation for later once we've got the UI down first. Since we tried tackling it all at once, we weren't very efficient with our time and it took a lot of time and effort and we were still not able to implement Flask right away. But we did get our UI implemented, and even though it was simple, it worked.

The last stretch of weeks we slowly upgraded our UI and added CRUD implementation to each of our tables. In addition to our CRUD and UI changes, we also added a new entity and implemented a dynamic search. The new entity was Factions, a group system that is attached to characters, but the field is NULLable. With the dynamic search, CRUD functions for every entity, and the dynamic search, we were able to complete the rest of our application and now it is fully functional.

Outline:

Users: records the details of the individual Account that each player has.

- userID: int, auto_increment, unique, not NULL, PK
- email: varchar(145), not NULL
- userName: varchar(25), not NULL
- Relationship: a 0:M relationship between Users and Characters is implemented using userID as the FK inside of Characters

Servers: records the details of the individual Characters that are hosted on the server.

- serverID: int, auto_increment, unique, not NULL, PK
- location: varchar(25), not NULL
- Relationship: a 0:M relationship between Characters and Servers is implemented using serverID as the FK inside of Characters

Characters: records the details of the characters that have been created by users.

- characterID: int, not NULL, PK
- characterName: varchar(16), not NULL
- level: int, not NULL
- userID: int, not NULL, FK
- serverID: int, not NULL, FK
- class: int, not NULL, FK
- Relationships: (1) a M:N relationship between Characters and Weapons is implemented using a connecting table Characters_has_Weapons where the characterID and weaponID are the FK; (2) 0:M relationship between Users and Characters is implemented using userID as the FK; (3) a 0:M relationship between Characters and Servers is implemented using serverID as the FK; (4) a 0:M relationship between Classes and Characters is implemented using classID as the FK; (5) a M:1 relationship with Factions using factionID as the FK.

Classes: records the details of the classes of characters

- classID: int, auto_increment, unique, not NULL, PK
- className: varchar(16), not NULL
- Relationship: a 0:M relationship between Classes and Characters is implemented using classID as the FK inside of Characters

Weapons: records the details of the weapons that are available to the characters within the game.

- weaponID: int, auto_increment, unique, not NULL, PK
- weaponName: varchar(50), not NULL

- damage: int, not NULL
- hit_pct: int, not NULL
- Relationship: (1) a M:N relationship between Characters and Weapons is implemented using a connecting table Characters_has_Weapons where the characterID and weaponID are the FK

Inventory: records the details of the weapons that are obtained by individual characters within the game.

- weaponID: int, not NULL, FK
- characterID: int, not NULL, FK
- Relationship: two 1:M relationships for Characters and Weapons since it is used as a composite entity to show the M:N relationship between Characters and Weapons. characterID and weaponID are the FK in this entity.

Factions: records the details of the weapons that are obtained by individual characters within the game.

- factionID: int, auto_increment, unique, not NULL, PK
- factionName: varchar(16), not NULL
- Relationship: a 1:M relationship with Characters.

1NF Check:

1. There are no repeating groups between each of the tables.
2. Each table has a primary key:
 - a. Users: userID
 - b. Servers: serverID
 - c. Characters: characterID
 - d. Classes: classID
 - e. Weapons: weaponID
 - f. Inventory: characterID, weaponID
 - g. Factions: factionID
3. The dependencies are identifiable:
 - a. Users: userID -> email, userName
 - b. Servers: serverID -> location
 - c. Characters: characterID -> characterName, level
 - d. Classes: classID -> className
 - e. Weapons: weaponID -> weaponName, damage, hit_pct
 - f. Factions: factionID ->

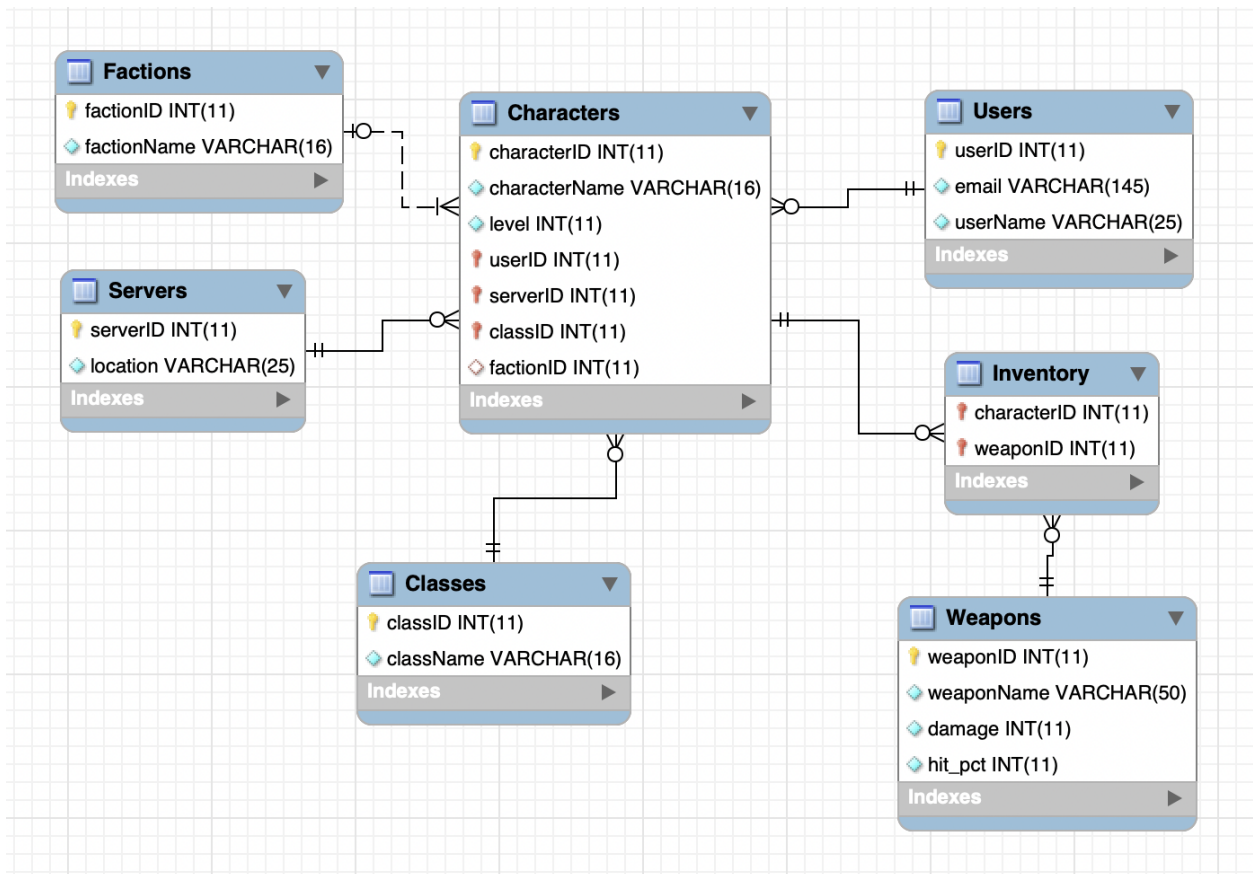
2NF Check:

1. Our tables have no partial dependencies and have been split to satisfy this condition.
2. Our tables are also in 1NF.

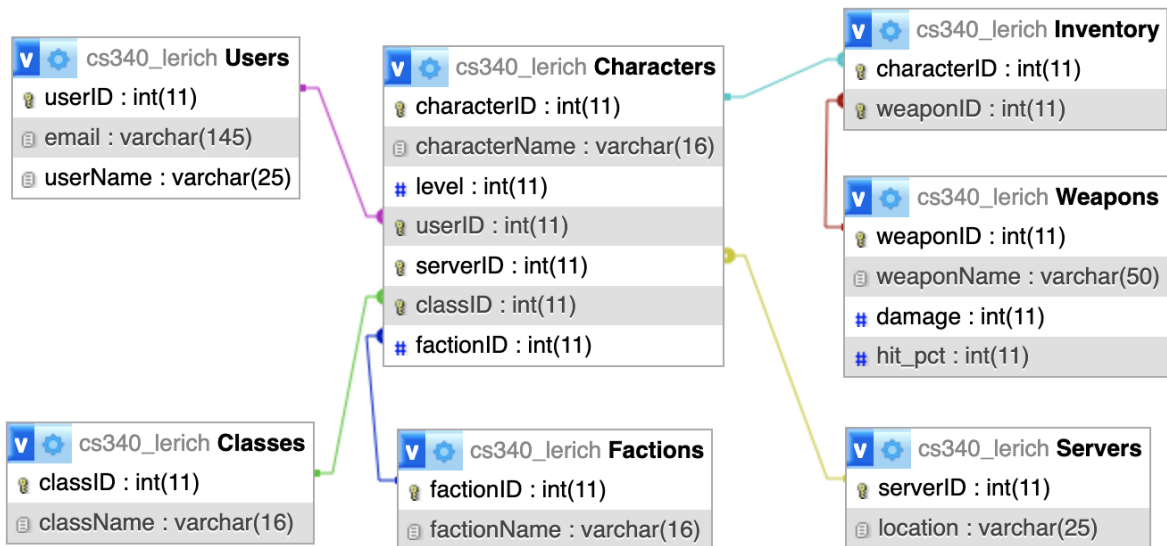
3NF Check:

1. Our tables have no transitive dependencies.
2. Our tables are also in 2NF.

Entity-Relationship Diagram:



Schema:



Example Data:

Users:		
userID	email	userName
1	joe12@aol.com	joecool
2	bill67@hotmail.com	bill_dance
3	tim2005@gmail.com	timturner
4	abi@icloud.com	abigail17
5	pat62@yahoo.com	patstar

Classes:	
classID	className
1	Warrior
2	Paladin
3	Mage
4	Priest
5	Druid

Weapons:

weaponID	weaponName	damage	hit_pct
1	Kang the Destoryer	100	75
2	Bo Staff	85	65
3	Excalibur	150	50
4	Durendal	75	83
5	Morning Star	62	100

Servers:

serverID	location
1	Bleeding Hollow
2	Lionheart
3	Aegwynn

Characters:

CharacterID	characterName	level	userID	serverID	classID
1	Merlin	16	1	1	5
2	Ivanoff	15	2	2	1
3	Arthur	18	3	2	2
4	Gandalf	21	2	2	3
5	Cristobal	10	4	1	4
6	Ragnar	9	5	1	1
7	Evander	17	4	3	1
8	Diante	14	5	3	2

Inventory:

characterID	weaponID
1	2
2	3
3	3
4	2
5	2
6	1
7	4
8	5

Factions:

factionID	factionName
1	Nuggets
2	Heat
3	BTS Army

UI Screenshots:

Create, Read, Delete User Page

[MAIN](#) [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Ogres And Alchemy

Users:

ID	Email	Name	Edit	Delete
1	joe12@aol.com	joecool	edit	delete
2	bill64@hotmail.com	bill_dance	edit	delete
3	tim2005@gmail.com	timturner	edit	delete
4	abi@icloud.com	abigail17	edit	delete
5	pat62@yahoo.com	patstar	edit	delete

Create User:

email:
Username:

Update User Page

MAIN [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Submit

Ogres And Alchemy

Edit User

ID

Email:

Username:

Save Changes

Create, Read, Delete Servers Page

MAIN [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Submit

Ogres And Alchemy

Users:

ID	Location	Edit	Delete
1	Bleeding Hollow	edit	delete
2	Lionheart	edit	delete
3	Aegwynn	edit	delete

Create Server:

Location:

Create Server

Update Servers Page

MAIN [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Submit

Ogres And Alchemy

serverIDLocation

serverID

Location:

Save Changes

Create, Read, Delete Weapons Page

[MAIN](#) [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Search Characters..

Submit

Ogres And Alchemy

Weapons:

ID	Name	Damage	Hit %	Edit	Delete
1	Kang the Destroyer	100	75	edit	delete
2	Bo Staff	85	65	edit	delete
3	Excalibur	150	50	edit	delete
4	Durendal	75	83	edit	delete
5	Morning Star	62	100	edit	delete

Create Weapon:

Name:

Damage:

Hit %:

Create Weapon

Update Weapons Page

[MAIN](#) [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Search Characters..

Submit

Ogres And Alchemy

Edit Weapon

ID

Name:

Damage:

Hit %:

Save Changes

Create, Read, Delete Classes Page

[MAIN](#) [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Search Characters..

Submit

Ogres And Alchemy

Classes:

ID	Name	Edit	Delete
1	Warrior	edit	delete
2	Paladin	edit	delete
3	Mage	edit	delete
4	Priest	edit	delete
5	Druid	edit	delete

Create Class:

Name:

Create Class

Update Classes Page

[MAIN](#) [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Search Characters..

Submit

Ogres And Alchemy

Edit Class

classID

className:

Save Changes

Create, Read, Delete Characters Page

Ogres And Alchemy

Characters:

ID	Name	Level	User	Server	Class	Faction	Edit	Delete
1	Merlin	16	joecool	Bleeding Hollow	Druid		edit	delete
2	Ivanoff	15	bill_dance	Lionheart	Warrior	Nuggets	edit	delete
3	Arthur	18	timturner	Lionheart	Paladin	Heat	edit	delete
4	Gandalf	21	bill_dance	Lionheart	Mage	Nuggets	edit	delete
5	Cristobal	10	abigail17	Bleeding Hollow	Priest		edit	delete
6	Ragnar	9	patstar	Bleeding Hollow	Warrior	Nuggets	edit	delete
7	Evander	17	abigail17	Aegwynn	Warrior	Heat	edit	delete
8	Diante	14	patstar	Aegwynn	Paladin	Heat	edit	delete

Create Character:

characterName:

level:

userID:

serverID:

classID:

factionID:

Update Characters Page

[MAIN](#) [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Ogres And Alchemy

Edit Character

characterID:

characterName:

level:

userID:

serverID:

classID:

factionID:

Create, Read, Delete Inventory Page

[MAIN](#) [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Search Characters..

Submit

Ogres And Alchemy

Inventory:

Character	Weapon	Delete
Merlin	Bo Staff	delete
Ivanoff	Excalibur	delete
Arthur	Excalibur	delete
Gandalf	Bo Staff	delete
Cristobal	Bo Staff	delete
Ragnar	Kang the Destroyer	delete
Evander	Durendal	delete
Diante	Morning Star	delete

Create Inventory Item:

Character:

Weapon:

Create, Read, Delete Factions Page

[MAIN](#) [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Search Characters..

Submit

Ogres And Alchemy

Factions:

ID	Name	Edit	Delete
1	Nuggets	edit	delete
2	Heat	edit	delete
3	BTS Army	edit	delete

Create Faction:

Name:

Create Faction

Update Factions Page

[MAIN](#) [USERS](#) [SERVERS](#) [WEAPONS](#) [CLASSES](#) [CHARACTERS](#) [INVENTORY](#) [FACTIONS](#)

Search Characters..

Submit

Ogres And Alchemy

Edit Faction

ID

Name:

Save Changes

Feedback:

New Feedback:

Review by Samuel Bervin:

Hi Backend Bros. aka Richard and John,

Please find my thoughts below. If anything in my review is unclear, please let me know. Good luck with the rest of your project! I think it's quite a cool one.

Best,
Sam

Does the UI utilize a SELECT for every table in the schema?

- Only two of the four tables were accessible. But it does do so for the two provided, and there's no reason to think that the two of you will have any issue finishing up the rest.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

- No, I don't believe so. I imagine that's in the pipeline, so good luck with it when you do add it in!

Does the UI implement an INSERT for every table in the schema?

- Same story as before: it does for USERS and SERVERS, but not for the other tables.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?

- I believe the plan is to do so with Characters:Weapons. And I do see instances in the SQL code like INSERT into the Characters entity using userID, serverID and classID as FKs. Hopefully it comes together!

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?

- Not as of now. You've got DELETE functionality for USERS and SERVERS, you just need to make sure you do this for your M:M relationship. (Characters:Weapons, correct?)

Is there at least one UPDATE for any one entity?

- Users and Servers both have UPDATE.

Is at least one relationship NULLable?

- This does not appear to be the case at this time. At least, I couldn't find anything in the SQL showing that. I was thinking of what you might do, and actually really like Riley's idea of weapons being NULLable for a character. A character starting unarmed is not rare, after all!

Do you have any other suggestions for the team to help with their HTML UI?

- I think this is quite a cool project, and would be interested in seeing how you put things together once you finish them. As it's not complete right now, I don't have too much to add here. Just stick to the more particular requirements (M:M-related, etc.), and you'll be golden! Good luck you two.

Review by Brian Anderson:

Hi Richard & John,

Interesting DB topic. I like the fact that you added a home page with some helpful explanatory test. Best wishes to you both on this assignment!

- *Does the UI utilize a SELECT for every table in the schema?* In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.
 - Not yet. I don't see any tables in the UI for weapons, classes, characters, and inventory.
 - You may not want to display all the UIs, for add, update, and delete, by default, on the users and servers page. However, I think it's useful to be able to do it if that is what the user wants to see.
- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
 - I see a SELECT in the DML.sql file that includes a JOIN and a WHERE clause to perform some filtering.
- *Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table.
 - I think the UI is missing the ability to insert into the weapons, classes, characters, and inventory tables.
- *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).
 - Yes, I think each INSERT adds corresponding FK attributes in the junction tables.

- *Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
 - Yes, I think I see that you used the CASCADE constraint in your DDL.sql so your DELETES should update linked records according to your CASCADE declarations.
- *Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
 - Yes, I see UPDATE queries for multiple tables in your DML.sql.
- *Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.
 - I don't see any defined fields that are allowed to be NULL. They all seem to be NOT NULL.
- *Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*
 - The obvious suggestion is to use 'AS' to create aliases that are easier for the reader to understand.
 - We added description on each web page that explained what the page was for, in case the user wasn't familiar with the data or the purpose of the table.

Review by Anthony Kowalski:

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

- Right now only the Users and Servers web pages are complete, it looks like the plan is to implement CRUD for all of the tables.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

- No, there is not currently one

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

- Same answer as earlier, there is only inserts for the Users and Servers pages.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products,

INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

- No, not yet

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

- There are deletes for the Users and Servers tables, the characters and weapons tables will have a M:M relationship utilizing a connecting table called Characters_has_Weapons, which still needs to be implemented

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

- Yes, there is an update

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

- I am not seeing one at the moment

Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.

- No, I think you guys are on the right track and just need some more time to finalize the front end. Good luck!

Review by Riley Rabelos:

Hi!

Your project is looking good so far! Your users and servers look great and seem to include all the necessary information. However, since your other pages are essentially blank you don't have all of the data included that you need. But I assume you plan on displaying the data similarly to how you have it displayed in the users and servers. If you do that it will be great!

You also need to make sure you include some sort of drop down menu or search mechanism in your final product.

Once again you guys included an insert for the first couple of tables but have nothing for the last few. If you continue with what you did for the first ones you would easily meet this requirement.

Since you don't have any tables implemented with foreign keys so far you also don't yet meet this requirement.

You do include deletes in your first couple of tables, however, you don't have a delete to remove something from a M:M relationship so you need to make sure that you include that with your finished product.

You definitely meet the update requirement since you include it in both of your finished tables. The update section is also very well formatted.

From what I can see you don't have any relationships that have the potential to be Null. You'll need to change this. For example you could make a characters weapon type Null if they do not carry a weapon.

I would maybe use the AS aliases to turn things like userName into just Username I feel like it would look cleaner. Otherwise, the layout of your pages looks very good so far. It's very clean and flows nicely. Once you finish adding the rest of the info you need it will look great! Good luck!

Old Feedback:

Review by Marcus Mcbay:

"Hello Richard and John,

I love this concept! Super curious how this will turn out, especially with implementing a server that different users can join. Maybe you can change the Characters_has_Weapons table to Inventory. What was the purpose behind naming "Characters_character, Users_user,... etc" like that? Overall, I think this project has a lot of potential.

Respectfully,

Marcus McBay"

Review by Benjamin Romaine:

"Hey Backend Bros!

Cool project! With something like this, it definitely seems like you could go down a rabbit hole of possibilities. It's a very creative way to do the assignment!

Your overview was great and gave a good description of the project. Since a game is more of a service and there isn't a direct problem to be solved, you did a good job listing the services offered to the user and I got a sense of the tables that would be used in the ERD.

The only thing I would mention is that in the overview, you say that a user can play with friends. So, maybe adding a Friends table that has a relationship with the User would include that feature of the game. A user could have a friends table, then if they are playing with friends you would have an intersection table with the user/character, friends, and the server IDs? A character could also then have a "squad" attribute? Just a thought! Like I said, you could go down a pretty big rabbit hole! But at the very least, including friends would be an important addition.

I agree with the previous comment about the naming conventions. They seem a little longer than they need to be and could be more concise. I think that will not only help cut down on the amount of typing (and possible typos that can occur), but can also add clarity to someone looking over your outline and ERD. I would also add that you have a "name" attribute in Users, Characters, Classes, and Weapons. To help with keeping things organized, I would suggest individualizing the attribute names like "user_name", "char_name", "class_name" and "weapon_name". With just "name" in there, things could get pretty confusing. I always like giving my characters regular names like Jeff. So knowing Jeff was a char_name versus a user_name would make sure there is no confusion.

Your ERD and relationships look good to me, my only notes are on the naming conventions and possibly showing that users have friends they can play with. And if they have friends, the relationships need to somehow connect the friends and the server they are playing on.

Great work!

Benjamin Romaine”

Review by David Emmons:

"Hi John and Richard, love the concept! Stray question, are either one of you working towards becoming a video game developer, maybe this would be a good starting point hahaha!

Looking at your overview, it is very clear and concise. You describe the requirements of the website well without needing to directly list them out. As mentioned above by Benjamin, the game is much more of a service than a "problem" right now, so I think the overview you have is great. More as food for thought, I think it would be interesting to think of possible problems in the future as well. Perhaps the game takes off and you need to solve a population issue to keep wait-time to get into servers down. Would you increase the number of servers directly, or just increase the number of people that can login to each server at once? How would that effect the way your backend is laid out?

This is where it might be useful to put some specific facts into the overview. Maybe mention how many servers you plan to have, the number of users that can be on each server at a given time, the number of characters each user can create on each server, etc. E.G. "Ogres and Alchemy is an MMORPG that is rising in notoriety. We currently have 10 unique servers, each server has a maximum userbase of 2000 people. There are plans to extend coverage to another region of the world, opening 5 new European servers with 2000 slots apiece.". Or talk about your class options "Each user can have 5 characters created per server, there are 15 classes to choose from ranging from Archer to Paladin, to Wizard."

You have met the guidelines for the number of entities, and each entity is its own singular idea. I think this would be a very good start to game design. There are obviously an infinite number of things you could add if you guys so chose to, friends list, mounts, items, skills, etc. But none are necessary and the current design you have is excellent.

All your 1:M relationships appear to be in order. You also have your M:M relationship in place between your characters and weapons. The only thing I would mention are the relationship between Characters and Classes, and Characters and Weapons. Are classes chosen during character creation? If so, then the ERD is perfect, if not, the relationship could be considered optional. The same goes for weapons, are weapons chosen during character creation? If so, is there any chance a character could lose their weapon / sell it and ever be unarmed. IF so this relationship should be optional as well :).

Lastly, your naming conventions all look good to me. Your tables are consistent using an uppercase first letter and being plural, whereas all the attributes are singular with lowercase starting letters, using camelCase to show multiple words.

Love to see more as you guys work on this!"

Review by Andrew Haylett:

"Hi Richard and John,

I love your idea of creating a character creation database for an MMORPG. So I enjoyed this one and look forward to seeing more from it.

Does the overview describe what problem is to be solved by a website with DB back end?

The overview does an excellent job of describing what the database will solve. This database will be used to house characters and what is associated with each character (Servers, classes, weapons, inventory, etc). I do think that you should add what problems your database will solve (i.e. Prevents characters from having the same name or having multiple servers prevents servers from being over crowded, etc). But besides that, I think that this overview was very well done.

Does the overview list specific facts?

The overview did list specific facts, it laid out what the role of this database was well and gave different objects that would be in the database. They even laid out requirements that this database would have telling that a character would need to be in the database before somebody could move to select a class. One thing that could be improved is to go into more detail on these things in your database. Maybe sharing how many maximum characters people can have or what your classes will be. It would help for you to get an even better idea of what you want this project to look like further down the line.

Are at least four entities described and does each one represent a single idea to be stored as a list?

You met the required number of entities, in fact, I was impressed with how much further you went adding even more than four entities. Good job on this part!

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Your outlines were well done, you shared what the entity did and how it related to other entities.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

All your relationships are correctly formatted and you have an M:M relationship between characters and weapons. Your ERD was very well done and presents a logical view of the database. I agree with the previous comment though on the fact that your relationship between weapons and character should be optional.

Is there consistency in

a) naming between overview and entity/attributes

Yes

b) entities plural, attributes singular

Yes

c) use of capitalization for naming?

Yes”

Actions:

New Actions:

1. Is at least one relationship NULLable? “This does not appear to be the case at this time. At least, I couldn't find anything in the SQL showing that. I was thinking of what you might do, and actually really like Riley's idea of weapons being NULLable for a character. A character starting unarmed is not rare, after all!” -Samuel Bervin

We added a factions entity that is a foreign key within the Characters entity. Factions is a Nullable field for a character.

2. “ You may not want to display all the UIs, for add, update, and delete, by default, on the users and servers page. However, I think it's useful to be able to do it if that is what the user wants to see.” – Brian Anderson

No action. We like that it's displayed up front for the database user to quickly access those functions. Adding an extra click to open a dropdown or menu would reduce the efficiency of the application as a user is most likely going to that page to make edits to the database.

3. I think the UI is missing the ability to insert into the weapons, classes, characters, and inventory tables. – Brian Anderson

No action. The “Add” functionality is an insert on each page/table.

4. I don't see any defined fields that are allowed to be NULL. They all seem to be NOT NULL. – Brian Anderson

Added a “faction” table and a M:M relationship in the character table.

5. The obvious suggestion is to use ‘AS’ to create aliases that are easier for the reader to understand. -Brian Anderson

Changed userName to username. The rest we feel like need to be fully spelled out like “characterName” doesn't need to be just “name” as that field name would be used in multiple tables if that were the case (e.g. weapon “name” and character “name” would have duplicate names when the tables are joined.

6. “I would maybe use the AS aliases to turn things like userName into just Username I feel like it would look cleaner. Otherwise, the layout of your pages looks very good so far. It's very clean and flows nicely. Once you finish adding the rest of the info you need it will look great! Good luck!” – Anthony Kowalski

Changed “userName” to ‘username’ in the Users table.

7. You also need to make sure you include some sort of drop down menu or search mechanism in your final product. - Riley Rabelos

Added a search mechanism on the for the “characters” and ”inventory” pages

Old Actions:

1. “Maybe mention how many servers you plan to have, the number of users that can be on each server at a given time, the number of characters each user can create on each server, etc.” - Marcus Mcbay

We added more numbers as facts in our overview to give a better picture of what we are trying to build our database for. This was a very helpful suggestion because Marcus also gave us ideas on what entities we should define rough numbers for.

2. “Are classes chosen during character creation? If so, then the ERD is perfect, if not, the relationship could be considered optional. The same goes for weapons, are weapons chosen during character creation? If so, is there any chance a character could lose their weapon / sell it and ever be unarmed. IF so this relationship should be optional as well :).” - David Emmons

David points out whether we want the relationship between characters and classes and characters and weapons to be optional and required. We do want classes to be set upon character creation so the relationship should stay the same, however we want weapons to be able to be picked up and dropped at will. So the relationship between characters and weapons will be changed so that it is optional.

3. “I would suggest individualizing the attribute names like "user_name", "char_name", "class_name" and "weapon_name".” - Benjamin Romaine

This is a good suggestion. We are going to change the naming to be more personalized for each entity.

4. “Maybe you can change the Characters_has_Weapons table to Inventory.” - Marcus Mcbay

We like this suggestion. It gives the name of the table more depth to it than something just literal.

Upgrades:

1. We are going to change the relationship between characters and weapons so that they don't require one another. Thanks to David Emmons for bringing this up.