Kyle Bell
Jeremy Tsang

Project Step 4 Draft Version: Design HTML Interface

(A) Project Outline and Database Outline

Project Title: Empire Employees
Team: Rakghoul Serum
URL: http://web.engr.oregonstate.edu/~tsangj/340/osu-cs-340-project/empire-employees/

Without proper organization and resolve the galaxy will turn to chaos, or worse, become under the control of rebel scum. Empire Employees manages over 1 billion Stormtroopers along with supporting droids and vehicles. To deploy a specific number of troops to ensure effective defense of each garrison spread such large cosmic distances it is critical to always have the correct number of supplies and personnel available. Empire Employees does this by allowing the users current troop, ship, and droid counts for each garrison and providing sufficient warnings when said counts drop below an acceptable level needed for adequate defense. For some ships, such as tie fighters, there may only be 1 trooper on board whereas for larger ships, such as star destroyers, may have as much as 9,700 Stormtroopers. With our simple to use online interface, we will meet all your Stormtrooper relocation management needs!

Entity Tables:

troopers - an individual soldier of the Galactic Empire that upholds peace and prosperity for all. They have a garrison and load out assignments that are related to the garrisons and loadouts entities respectively.

- id: int, auto_increment, unique, not NULL, PK
- garrison: int, FK
- loadout: int, not NULL, FK
- Relationship: a 1:M relationship between garrisons and troopers, a M:M relationship between troopers and ships (implemented with the ships_troopers composite entity), and a 1:M relationship between loadouts and troopers.

ships - The Galactic Empire has ships in order to transport troopers and droids throughout the galax. Each ship has a ship type and is related to troopers and droids.

- id: int, auto_increment, unique, not NULL, PK
- type: varchar(255), not NULL
- Relationship: a M:M relationship between ships and troopers, a M:M relationship between ships and droids.

droids - Robust, top of the line droids help repair ships for our glorious fleet. They have a droid type, and have a relationship with ships.

- id: int, auto_increment, unique, not NULL, PK
- type: varchar(255), not NULL

Kyle Bell
Jeremy Tsang

- Relationship: a M:M relationship between droids and ships implemented with the ships_droids composite entity.

loadouts - Each unique weapon loadout contains a proprietary blaster as well as an option for a detonator in order for each trooper to get the job done right their own way! A loadout has a relationship with a trooper.

- id: int, auto_increment, unique, not NULL, PK
- blaster: varchar(255), not NULL
- detonator: bool, not NULL
- Relationship: a 1:M relationship between loadouts and trooper

garrisons - Our Empire keeps the galaxy safe with garrisons throughout the galaxy. Each of them has a name and a maximum capacity

- id: int, auto_increment, unique, not NULL, PK
- Name: varchar(255), not NULL
- capacity: int, not NULL
- Relationship: a 1:M relationship is implemented between troops and garrisons as a FK inside of troops.

Relationship Tables:

ships_troopers - Relationship between ships and troopers.

- ship: int, not NULL, PK, FK
- trooper: int, not NULL, PK, FK
- Relationship: facilitates M:M relationship between ships and troopers.

ships_droids - Relationship between ships and droids.

- ship: int, not NULL, PK, FK
- droid: int, not NULL, PK, FK
- Relationship: facilitates M:M relationship between ships and droids.

Team Assignments:

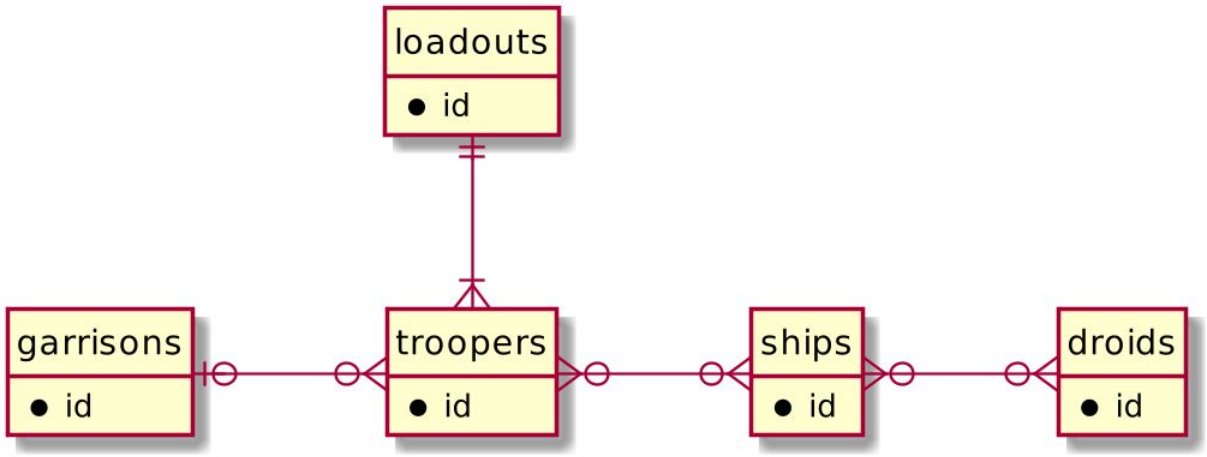
We will be implementing all of the entities described above:

- troopers
- loadouts
- garrisons
- ships
- droids
- ships_droids
- ships_troopers

- Kyle will be responsible for the code and webpages for the following entities:

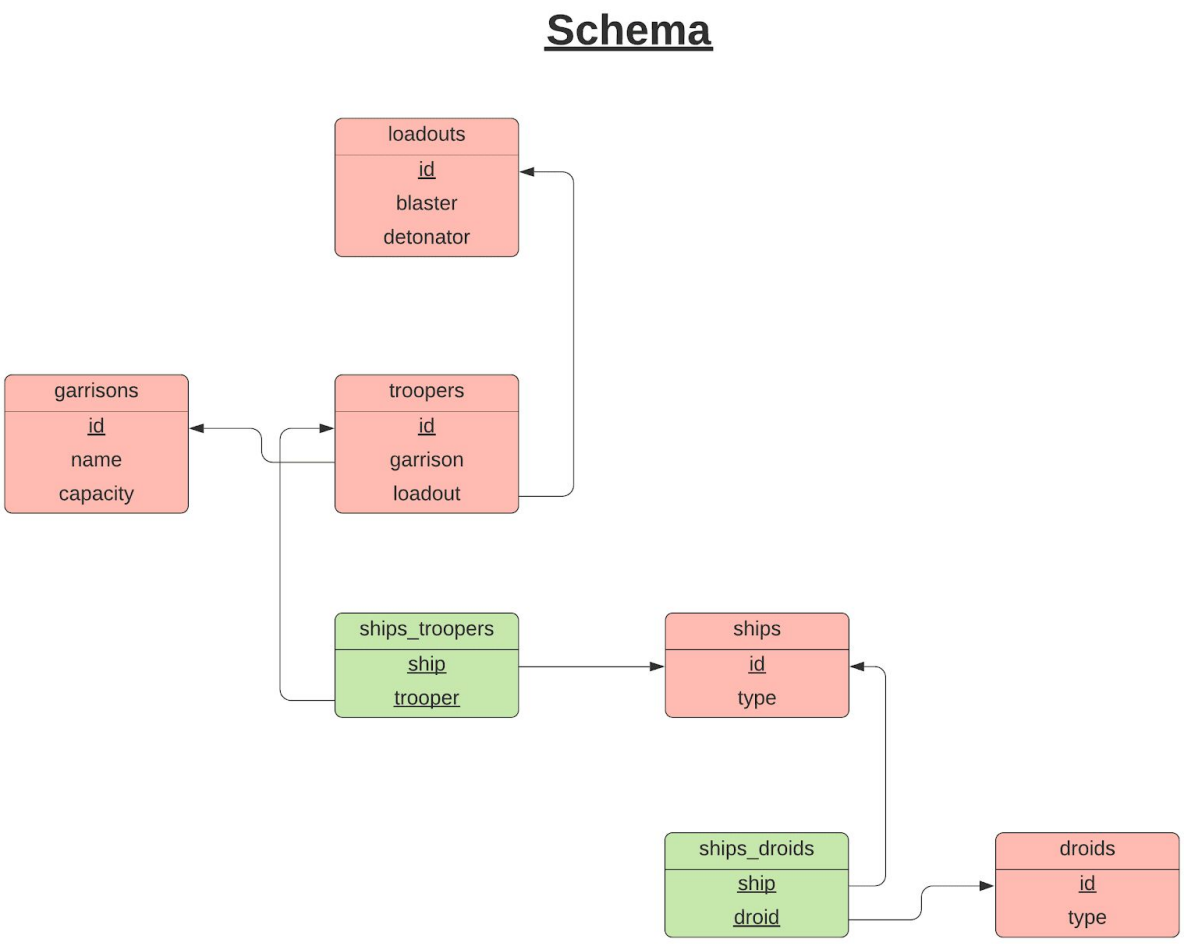
    - ships_droids (M:M relationship)
    - droids
    - garrisons
    - ships
- Jeremy will be responsible for code and webpages for the following entities:
    - ships_troopers (M:M relationship)
    - troopers
    - loadouts

(C) Entity-Relationship Diagram



Kyle Bell
Jeremy Tsang

(D) Schema



Kyle Bell
Jeremy Tsang

(B) Fixes based on Feedback from Previous Steps

Actions taken from feedback by the peer reviewers From Step 3 Draft

- We did not images, but we will add images afterwards [Brian Peck]
- trooper id to garrison id on Garrison page [Caroline Borden]
- garrison column names [Caroline Borden]
- loadouts column names [Caroline Borden]
- Clarify 2nd loadout form as “update” instead of “add” [Brian Peck, Caroline Borden]
- suggestion: remove a trooper all together (if they die) [Caroline Borden]
- Make the tables larger to fill up whitespace [Anjelica Benitez]

Feedback by the peer reviewers From Step 3 Draft

Kyle Bell
Jeremy Tsang

Kyle Bell 6 days ago
Team (64): Rakghoul Serum
Project Title: Empire Employees
Team Members: Jeremy Tsang, Kyle Bell

URL: **http://web.engr.oregonstate.edu/~tsangj/340/osu-cs-340-project/empire-employees/**

group64_step3_draft.zip

helpful! | 0

Brian Peck 6 days ago

- 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.
  - Yes it does! There are individual pages for each table and data is displayed from a select statement on each one of them. Looks good!
- Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
  - Yep! The troopers page offers this functionality.
- 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.
  - They do! Each individual page has the option to add rows to the table and the data being inserted matches the column headers.
- 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).
  - Many to many relationships exist between ships and troopers and ships and droids. There is a Manifests page that would allow a user to add a ship and it's occupants to it. This satisfies the request to add rows to the intersection table.
- 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.
  - The DELETE option exists in the Manifests page as well. It looks like you are able to remove the relationship between a ship and it's troop. I would assume the way this will be programmed will not remove the values from the ships, troopers, or droids tables. Looks good!
- 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?
  - The home page specifies the UPDATE functionality will be included on the Loadouts page, but I don't actually see it when I go to the page. I see two options to ADD. I think the second box is probably meant to say UPDATE since it includes an id, but that is something to look at.
- 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.
  - Yep! This requirement is satisfied on the Troopers page where you can update the troopers garrison to be NULL.
- Do you have any other suggestions for the team to help with their HTML UI?
  - Looks like you're off to a great start! I think the idea of breaking it up into an individual page for each table is a smart idea and made it easy for me to understand what your project was trying to achieve
  - Adding some images could help enhance some of the pages. I do like the color scheme though. Well done!

helpful! | 0

Kyle Bell
Jeremy Tsang

Caroline Borden 4 days ago

- *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.

Yes, this information is explained on the home page and implemented on the additional pages by displaying a table. It is very clearly and effectively demonstrated that there can/will be a SELECT query for each table.

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

Yes, the Troopers page utilizes a search/filter based on the current garrison assignment/loadout and said Trooper can be removed from the garrison or added to a new garrison as needed.

- *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.

Yes, this information is explained on the home page like with the SELECT queries and implemented on the additional pages by displaying a table that can be updated as items are added. The Garrisons page appears to be incorrect with the current options, though. I suspect that it's still being worked on, but I would expect the add options to be garrison id, name, and capacity instead of trooper id and garrison – similar to how the ships and droids pages are implemented.

- *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, where necessary there appears to be a reference to the FK attributes that are needed.

- *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, there will be multiple DELETE options. Like stated before, a trooper can be removed from a garrison or moved to a garrison on the Troopers page.

- *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, again the Troopers page is an excellent example of all of these queries and a Trooper can be updated to a new garrison on this page.

- *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.

Yes, the Trooper can be removed from a garrison and the table allows this to be set to NULL.

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

I would suggest maybe having an option to delete a Trooper altogether – should they dutifully give their lives in service to the Empire. I would also update the garrisons form/table as mentioned in the INSERT comment to reflect the garrison's attributes. The loadouts insertion seems confusing to me as well. There is an option to add blaster/detonator and a separate option to add id/blaster/detonator. Do these relate to different additions or is it an accidental duplicate? Also the table on the page references name/capacity. I would suggest having these reference blaster and detonator to be clear what is being displayed.

Overall, I think you've done a commendable job in your efforts to further the Empire's inevitable success to bring order to the galaxy.

(Seriously, I love the Star Wars theme and I'm excited to see the final result!)

helpful! | 0

Kyle Bell
Jeremy Tsang

Anjelica Benitez 4 days ago

- ***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.**
  - Yes, each table utilizes a SELECT for every table in the schema as outlined in the home page.
- ***Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?***
  - Yes, the Troopers page allows a user to filter by Trooper ID, Garrison, or Ship.
- ***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.**
  - Yes, INSERT is implemented in every table using the "Add" function.
- ***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, there are M:M relationships between Ships and Droids and Ships and Troopers. INSERTing a new row in either of these entities would also insert a new row in the corresponding relationship table.
- ***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, DELETE is implemented in the Troopers page. Also, the home page indicates that you can DELETE things from M:M relationships in the Manifests page.
- ***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, according to the home page, the Loadouts table will offer an UPDATE funtionality.
- ***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.**
  - Yes, the Troopers table allows a user to input NULL in the garrison field to remove a 1:M relationship.
- ***Do you have any other suggestions for the team to help with their HTML UI?***
  - What an awesome website! I love the layout of the website and how detailed you were with the home page. The only minor suggestion I can think of would be to maybe increase the size of your tables as there is a lot of white space. Overall, though, great job!

helpful! | 0

Kyle Bell
Jeremy Tsang

Zan Zhang 2 days ago

- *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.

Yes it provide the place where set in each entity on homepage with the SELECT to display entire table message.

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

Yes in Troopers there has filters on garrison and trooper id.

- *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.

Yes, in each individual page has the button to add the data in to the table.

- *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, the web has a page under COMPOSITES part which shows the M:M relationships between Ships and Droids and Ships and Troopers, and now it has the add on it. It seems correct for now.

- *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 they have delete message on the Manifests page to delete the M:M relationship. So it seems like they are good now.

- *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 base on the home page and each entities page. They have at least one UPDATE for each part.

- *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.

Yes, the table allows this to be set to NULL when the Trooper is removed from a garrison

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

The website setup very clearly and I really like the homepage which you list each function in and this will give a very clear instructions for the people who use the system you write. And I think you are the fewer team do this on the homepage. I recommend remain this until the end.

helpful! | 0

Kyle Bell
Jeremy Tsang

Actions taken based on feedback from submitting Step 2 Final Draft

No feedback was received from grades after submitting Step 2 Final Draft

Actions based on the peer feedback for step 2 draft review

- Per Rachel Orrell's suggestion, updated the schema reviewer to simply use arrows instead of crow's feet notation. This makes it a bit more streamlined and gets rid of extra information that is included in the ERD anyways.
- Per Rachel Orrell's suggestion, specified that the loadout is a weapon loadout in the outline to be clear that the loadout is weaponry
- Per Rachel Orrell's suggestion, fixed the trooper loadout relationship to consistently be 1:M. This is in accordance with the fact that a loadout can be shared among troopers.
- Per Rachel Orrell's suggestion, reworded/cut down the last outline sentence for better readability
- Per Rachel Orrell's suggestion, removed "_id" from ships_id, trooper_id, and droid_id for consistency and readability.

Actions not taken based on the peer feedback

- We did not implement separate entities for blasters and detonators as suggested by Jordan Pemberton. Those are implemented by our sister company "Empire Weapons."
- We did not include more attributes for each entity (suggested by Dylan Albertazzi) at this time to ensure we have a base database that functions well. As the project progresses we will consider adding more attributes.

Feedback by the peer reviewers For Step 2 Draft

Jordan Pemberton

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

Yes.

Does the overview list specific facts?

Yes, stormtrooper numbers, different numbers of troopers /ship type, etc.

Kyle Bell
Jeremy Tsang

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

Yes. Could maybe include blasters /detonator as separate entities from loadouts.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

Yes.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Yes, very consistent.

Rachel Orrell

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

Yes, and the idea is very fun! Love it!

Does the overview list specific facts?

Yes, the number of Stormtroopers managed and the number of troopers that can be supported by certain ships.

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

There are five entities, and I think so, though I don't really understand what a loadout is.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

The purpose of a loadout is not really clear. It says it helps troopers get the job done right - what job? Is it a weapon? I'm just not sure. Also, the description makes it sound like 1:1 ("*a*

Kyle Bell
Jeremy Tsang

loadout has a relationship with *a* trooper") while the relationship is listed as M:M. Looks like this was recently changed to 1:M, so neither one really fits. Finally, at least how it is now, shouldn't loadout be specified as an FK?

Otherwise looks good!

It is clear who is responsible for what.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

The implementation of the 1:M for loadouts and troopers makes it so that one loadout can belong to many troopers. Since I don't really understand what a loadout is, I could be off base on this, but if it is a kind of weapon, wouldn't it be more logical for a trooper to have multiple loadouts (the trooper id in the loadout table instead of the loadout id in the trooper table)?

There are 2 M:M relationships, and they are correctly formulated.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Almost. The foreign keys are referred to as something_id in the relationship tables, but foreign keys in entity tables are just referred to by the singular of the name of the reference table. Having all foreign keys end in "_id" would make it more obvious when an attribute is a foreign key and would improve consistency.

Notes: you don't actually need the crow's feet notation in the schema, and trying to put it there is really confusing when it comes to relationship tables. For example, there will likely be more than just one trooper in ships_troopers, but the crow's feet make it look like exactly one.

Dylan Albertazzi

Does the overview describe what problem is to be solved by a website with DB back end?
Great, you stated the problem, how a DB will help, and some specific details. I would reword the last sentence of your outline paragraph, or split it into two sentences.

Does the overview list specific facts?

Good job listing the specific number of Stormtroopers.

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

Kyle Bell
Jeremy Tsang

Yes, five entities are described. I would encourage you to think about more data on each entity that would be useful for your users to have. Such as date the ship was bought or an hourly wage of a trooper.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

Yes, you did a good job being brief but descriptive. I appreciate that. Your descriptions sound a bit like a sales pitch, which is fine, but you're selling the software, not the entities.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes, all relationships are formulated correctly. Good job.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

All naming is consistent with. Good job.

Patrick Dougan

1. The overview describes the challenges of managing a galactic army. It also explains how the DB will assist with managing troop counts and checking what ships troops are stationed on.

2. The overview lists a couple of facts. One is that total troop count is over 1 billion. Two is star destroyers hold between 1 and 9,700 troops.

3. Yes, they have five entities. Troopers, ships, droids, load outs, and garrisons. Each entity can be described by a list.

4. All entities are described with datatypes and constraints. Relationships are described as well.

5. There are two M:M relationships. The 1:M are correctly formulated

6. Yes, all entities are plural. The "Name" attribute is capitalized in the entity tables list but is OK in the schema.

Upgrades to the Draft version

No upgrades made on our own after submitting the draft. Upgrades based on the peer comments were sufficient and our database was already in 3NF.

Kyle Bell
Jeremy Tsang

Fixes based on Feedback from Step 1

We did not receive any feedback but we made the following changes:

- Switch naming conventions from camel case to underscores.
- Rename the composite entities shipTrooperManifest and shipDroidManifest to ships_troopers and ships_droids respectively.
- Change loadOuts to loadouts since loadout is a single word.
- Change the relationship between loadouts and troopers from M:M to M:1.
- In the Database Outline for ship_troopers indicate that together ship and trooper form the primary key for composite entity ship_troopers.
- In the Database outline for ship_droids indicate that together ship and droid form the primary key for composite entity ship_droids.