Carrie Davis
Sarah Turner
Group 15: “Rainbow Sparkles”
CS 340 Section 400/401

Project Step 4 DRAFT Version: CRUD

URL: http://flip2.engr.oregonstate.edu:6332 with VPN

Feedback by Peer Reviews

STEP2

Review 1:

- *Note to Group15: Did you accidentally upload last week's step1 document?

ERD

1. ... 5.I think the team forgot to add their ERD, theres only a database outline. Though its very thorough and meets the needs of a schema definition, but could used more explicit identification of integer foreign keys.

Schema

1. ... 4.Team's design document left out the schema diagram. This would have been useful in determine the integer foreign key references between the entities.
2.

- What's up Rainbow Sparkles team! Loving the theme :D

Review 2:

The best peer review for ERD would answer all of the following questions.

1. Are the attributes for each entity in the ERD same as that described in the database outline?
    1. I don't see ERD. The description works it out in depth though. I'll update review when group 15 updates with an ERD :)
2. Is the participation of entities in the relationships same as that described in the outline?
    1. N/A
3. Is the cardinality of entities in the relationships same as that described in the outline?
    1. The relationships in the database are worked out in words and looks like they've accounted for each of the different relationships.
4. Based on the Database outline, could any of the relationships be better off described as an Entity instead ?
    1. From the description, I don't think any relationships need to be their own entities.
5. Is there something that could be changed/improved in the E R Diagram and/or the overall database design?

    1. Just add an ERD :) also schema :)

The best peer review for a Schema would answer all of the following questions:

1. Are the relationship tables present where required and correctly defined, when compared with the database outline?
    1. Schema hasn't been uploaded yet.
2. Are foreign keys present where required and correctly defined, when compared with the database outline?
    1. I'm sure they will be. They've highlighted the primary keys in the database outline, so that's a start.
3. Do the entity attributes match those described in the outline?
    1. We'll see when schema is uploaded.
4. Is there something that could be changed/improved in the Schema and/or the overall database design?
    1. Just upload it and i'll re-review

Review 3:

ERD

Are the attributes for each entity in the ERD same as that described in the database outline?

- Yes, for the most part the attributes are the same
- Not sure if this is an issue or not, but I noticed that there are a couple of minor naming inconsistencies between the description and the ERD that made it slightly more difficult to follow, for example:
    - Description says that Groups have a City_id characteristic, but ERD says that Groups have "headquarters"
    - Description says that Characters have Type_id and Group_id, but ERD says that Characters have "type" and "group"

Is the participation of entities in the relationships same as that described in the outline?

- Yes, this seemed consistent
- The occupation/type relationship is very interesting! I think I understand that this relationship only exists if the occupation is type-exclusive... otherwise there's no relationship between occupation and type. Is this correct? If yes, some additional explanatory text in the relationships bullet section would be helpful. (Er, actually, if I'm wrong, some additional explanatory text would still be helpful)

Is the cardinality of entities in the relationships same as that described in the outline?

- For the most part, yes, but there were a few differences in what I imagined after reading the outline, and what I saw in the actual ERD:
    - I hadn't realized that characters MUST have a city (it's a "1 1" in the ERD, and I'd assumed it was going to be an " 0 1" relationship). Are any ponies from the country or are they all urban-dwellers? :)

    - Must a character be in a group? The description says that the group_id "can be null", but the ERD shows that the character has a "1 1" relationship with group. Should this be "0 1"?
- For clarity, it would be good to consistently include mandatory/optional information in the bulleted section of the outline so that it's clear which attributes are required and which can be null.

Based on the Database outline, could any of the relationships be better off described as an Entity instead ?

- I don't think so, it seems like the relationships and entities noted in the ERD are consistent with what's in the outline.

Is there something that could be changed/improved in the ER Diagram and/or the overall database design?

- The primary key notation in this ER is a bit different from the Week 2 notation (an asterisk).
- Design-wise, could you clarify the relationships between characters, cities, and groups? The ERD implies that each character must have a city, and that each character must have a group. But groups MIGHT have a city. What happens if a group is based in City A, and a character is part of the group but lives in City B. Does the character simply commute?
- The description for City name says "This will be the name of the family or group", which doesn't really describe a city. Possibly a copy-paste error?

Schema

Are the relationship tables present where required and correctly defined, when compared with the database outline?

- Yes!
- Also I like that there are titles for Entities and Relationships. That's not mentioned as part of the Week 2 notation that we're meant to use, but I found it quite helpful

Are foreign keys present where required and correctly defined, when compared with the database outline?

- The foreign keys are missing the connecting arrows that show where the data comes from (for example, in the Character table, Type_ID should have an arrow that connects it to the primary key of Type)

Do the entity attributes match those described in the outline?

- Yes!

Is there something that could be changed/improved in the Schema and/or the overall database design?

- Adding arrows for foreign keys

- Including adding arrows for any attributes in the relationship tables, since those are all foreign keys

Good job guys, I hope I get to make a pony…

Review 4:

ERD

1. Are the attributes for each entity in the ERD same as that described in the database outline? Yes. I think basically the description matches the ERD diagram.
2. Is the participation of entities in the relationships same as that described in the outline? Yes.
3. Is the cardinality of entities in the relationships same as that described in the outline? Yes.
4. Based on the Database outline, could any of the relationships be better off described as an Entity instead ? I think this group has done a good job.
5. Is there something that could be changed/improved in the ER Diagram and/or the overall database design?

I think this group has done a good job. I don't think it needs to be modified.

Schema

1. Are the relationship tables present where required and correctly defined, when compared with the database outline? Yes. The table correctly defined with the database outline.
2. Are foreign keys present where required and correctly defined, when compared with the database outline? Yes.
3. Do the entity attributes match those described in the outline? Yes.
4. Is there something that could be changed/improved in the Schema and/or the overall database design? I think this group has done a good job, and it is very clear. I saw the Shila advice that adds arrows. I am not sure is it necessary, because I didn't add too.

STEP3

PEER REVIEWS

Review 1:

….2. Is there a better way that data could be displayed on SHOW functionality pages?

I really enjoy the ease of use to add and update the character! I assume the delete function has not been implemented yet. I am looking forward to test it out once it has been added to the webpage.

...

4. Is there a better way that the DELETE functionalities could be implemented?

It has not been implemented yet but if it is similar to the quality of add/update character, it will be amazing!

5. Is there a way to search OR filter data ? (Remember, it need not work since these are just HTML pages)

It allows me to filter the ponies by group, type, city, and occupation(job). Overall, I really enjoy the layout and functionalities of the webpage!!!

DDQ

1. Is the SQL file syntactically correct?

I got an error from importing the ddq because DROP TABLE is specific to your database, so I had to change the code a bit in order to import. Afterward, I got SQL query error for CREATE TABLE `Character. It says incorrect table definition.

#1075 - Incorrect table definition; there can be only one auto column and it must be defined as a key

Review 2:

1. Does each functionality listed in the CS340 Project Guide have a corresponding HTML page? (*It's okay to implement multiple functionalities on the same HTML page*) —> 's a little confusing as to what functionality is included. There is a place to submit on each page but i'm not sure if this is for insert or update or both. Also there is no delete and the search is a little confusing….

   DDQ

1. Is the SQL file syntactically correct? there are some errors, i think from reading the post above it is about having the database be specific to you.

a) Actions based on Peer Feedback :

STEP 1

- Changed character to group relationship from 'one to one' to 'zero to one' and also changed group to character from 'zero to many' to 'one to many'. This makes sense as not all characters have a group and all groups have at least one character.

- Changed attributes and entity name for more consistent language, for attributes using 'city' instead of 'headquarters' and 'job' instead of 'occupation' in entity.

- Updated foreign keys Character.type_id, Character.group_id in the Character ERD to match description.

- Fixed typo in the name attribute for city.

- Changed type.magic and type.flight Type.equestrian and job.type_exclusive from Yes/No to BIT based on instructor's comment " While your implementation for booleans is fine, I'd recommend using a BIT value which can be either 0, 1, or NULL."

- Updated description in type exclusive relationship to better explain based on feedback.

- In reference to the feedback "I hadn't realized that characters MUST have a city (it's a "1 1" in the ERD, and I'd assumed it was going to be an " 0 1" relationship). Are any ponies from the country or are they all urban-dwellers? :)" the character description attribute city was updated to allow for a default of Unknown. However, we didn't change the relationship to be 1 to zero because surprisingly all the characters as far as we've seen have a city associated with them.

- Updated ERD based on comments on the job relationship to type. To make a zero to one relationship.

- Changed schema to be represented by tables and use connectors with arrows to represent one to many relationship.

STEP 2

- Removed the photo_id attribute as it made more sense to use the id anyways.

b) Upgrades to the Draft Version

- Made type to job a 'zero to one' vice a 'one to one' relationship to be a more accurate reflection of the description and our intended implementation.

- Removed comment on ERD after revising description for type-group entities and relationship.

- Reordered schema diagram to make arrows easier to follow

- Made 'cities' into 'city' since entity is singular
- Modified names of some variables based on ease when writing sql queries
- Added some delete restrictions based on foreign keys for character foreign keys (you can't delete a TYPE linked to a character, etc)
- Removed photo_id as an attribute.
- Made all BIT fields BOOLEAN because it made the JSON easier to parse
- Made changes to our DML based on our front end design. This included adding better filter queries for the character entity, using LEFT JOINs vice INNER JOINs, and removing deletes from everything but the character entity and character_job relationship

c) Project Outline and Database Outline - Updated Version:

Project Outline

This database maps out the characters of the My Little Pony Friendship is Magic Television Series that aired from 2010 - 2014. Wiki page: https://en.wikipedia.org/wiki/List_of_My_Little_Pony:_Friendship_Is_Magic_characters . Fandom page: https://mlp.fandom.com/wiki/My_Little_Pony_Friendship_is_Magic_Wiki The focus is on this particular cast of fun and cute characters. In this series the show focuses on diverse personalities with complex relatable issues. These ponies come from different groups, cities in different regions, and can be of different types. Think of a child friendly, relatable game of thrones minus genocide. Many characters have occupations or multiple occupations throughout the Equestria empire. This is all flexible as the fun drama of pony life plays out. The character list is almost exhaustible and this database will allow users to add characters to complete the character list. The database will allow characters to be filtered by group or type.
The complexity and ability to build relationships between entities makes this a good candidate for a database project.

Database Outline, in Words

The entities in this database are:
Focused on the My Little Pony series Friendship is Magic.

- Character -The primary entity in the database and participates in a relationship with every other entity in the database. Each one has the following attributes:
    - * id(int): Unique Int field that will be primary key that auto increments and cannot be NULL.
    - name(varchar, length: 100): Text name of character that can be up to 100 characters long. This cannot be NULL and there is no default.
    - type_id: an integer that acts as a foreign key to the type entity. This can be NULL.
    - group_id: an integer that acts as a foreign key to the group entity. This can be NULL.
    - gender(ENUM): Character field describes gender. “M, F, O (non-binary)”. Cannot be NULL.
    - city_id: an integer that acts as a foreign key to the city entity. This can be NULL.

- Type - Type entity identifies the species of characters on the series. A type can share characteristics and are differing in characteristics from other types. Each character has a type that can be shared among multiple characters.
    - * id(int): Unique Int field that will be primary key that auto increments and will not be NULL.
    - type_name(varchar, length: 25): Text name of type that can be up to 25 characters long. This can not be NULL and there is no default.
    - flight(bit,length:1): BOOLEAN field which value represents if type can fly or cannot fly. 0 will represent No and 1 will represent Yes. NULL will mean unknown.
    - magic(BOOLEAN, length:1): BOOLEAN field which value represents if type can perform magic or not. 0 will represent No and 1 will represent Yes. NULL will mean unknown.

    - equestrian(BOOLEAN,length:1): BOOLEAN field which value represents equestrian or not equestrian. 0 will represent No and 1 will represent Yes. NULL will mean unknown.

- City:
    - * id(int): Unique Int field that will be primary key that auto increments and will not be NULL.
    - city_name(varchar, length: 100): This will be the name of the city. Each must be individual/unique, and cannot be NULL. Default will be 'Unknown'.
    - characteristics(varchar, length: 256): Text description of the city that can be up to 256 characters long. Default is NULL.

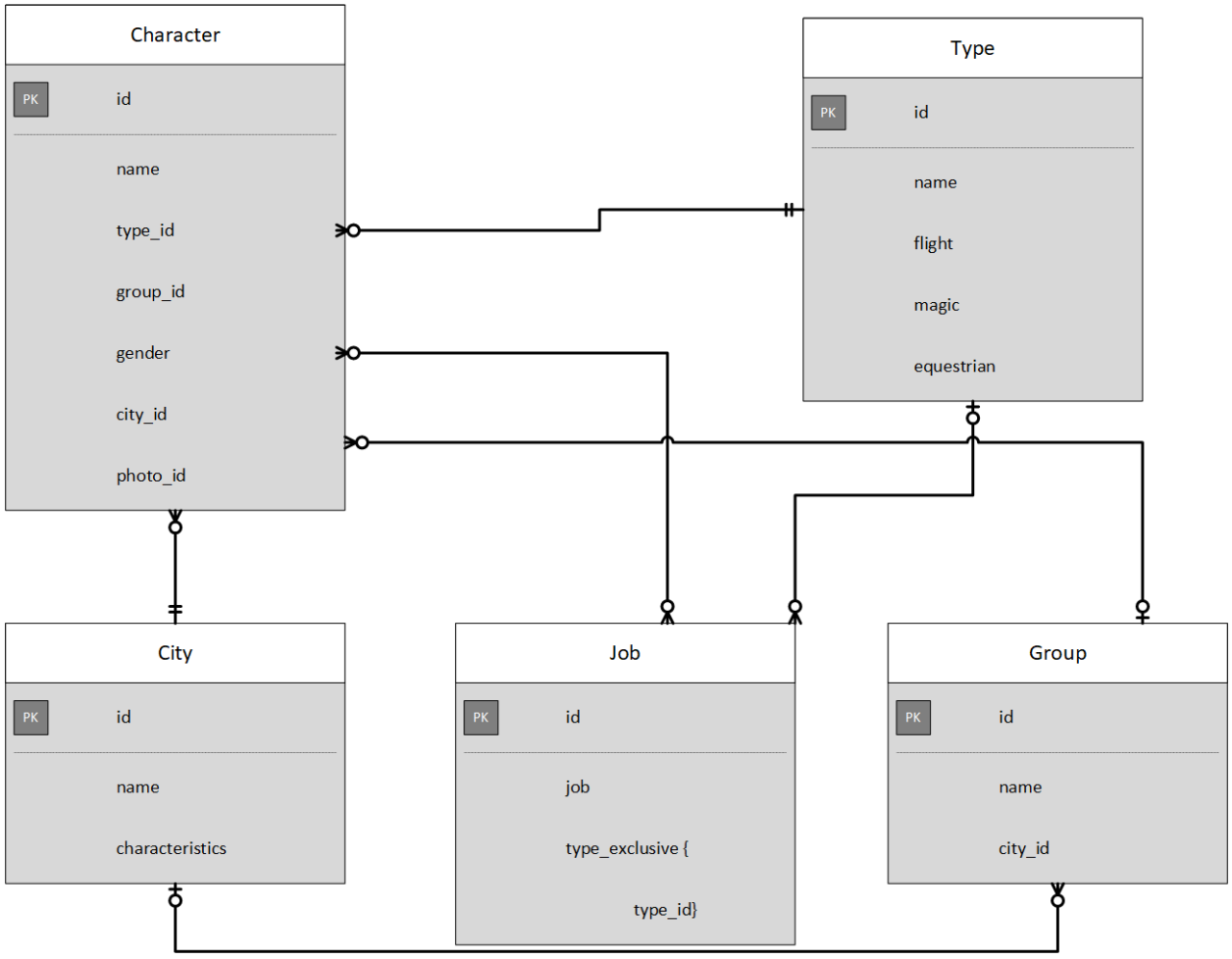
- Group - The characters are grouped in the series. Each group has a zero to many relationship with the characters.
    - * id(int): Unique Int field that will be primary key that auto increments and will not be NULL.
    - group_name(varchar, length: 100): This will be the name of the family or group, 100 character maximum. Each must be individual/unique, and cannot be NULL. Default will be 'Unknown'.
    - city_id: An integer that acts as a foreign key to the city entity. Can be NULL.

- Job: Characters can have multiple jobs throughout the series or no job.
    - * id(int): Unique Int field that will be primary key that auto increments and will not be null.
    - job_name(varchar, length: 100): Text name of the occupation with 100 character maximum. Each must be individual/unique, and cannot be null. Default will be unknown.
    - type_exclusive(BOOLEAN,1): type_exclusive field describes BOOLEAN value for whether a job can only be done by a particular type of character or not. 0 will represent No and 1 will represent Yes. NULL will mean unknown.
        - type_id(int): If 1 for type_exclusive, then this attribute will contain an integer that acts as a foreign key to the type entity. Default is NULL.

The relationships in the database are:

- **Characters are from cities** - There are many cities, but a character can only be from one. Multiple characters can be from one city. One-to-many relationship.
- **Characters have a group -** Characters can be part of one group. Characters can only have one group. One-to-many relationship.
- **Characters have jobs -** Some may have multiple jobs and some may not have any. Characters can share the same occupation or have different ones. Many-to-many relationship.
- **Characters are of one type**- Characters are one type, but multiple characters can be of that species. One-to-many relationship.
- **Groups are from a city** – groups can have a headquarters (city) or not. Many groups can be from one city but each group can have at most one headquarters. One-to-many relationship.
- **Some jobs can be performed by only one type of character** – If the job.type_exculsive value is 1 to represent "Yes" then the type_id will be set to represent the type of character that can perform the job. The job to type relationship can be zero or one.
- **Types of characters can have zero or many type exclusive jobs -** The type.id can be linked to many different jobs or none. For example. A "Pegasus" type can have multiple different jobs often flight or weather related (eg. flight Instructor, weather patrol).

d) Entity-Relationship Diagram:



e) Schema:

Relationships