Project Step 4 Draft Version: CREATE & READ Operations

Danny Tran and Jesse Madrigales

URL: http://flip1.engr.oregonstate.edu:3031/

a) Project Outline and Database Outline, ERD and Schema Updated Version

Project Outline

We will be attempting to build a database to represent the Game of Thrones **show** Universe**.** The show is based on series of fantasy books by author George RR Martin. It is set on the fictional continents of Westeros, Essos and Sothoryos (rarely mentioned but mentioned none the less). To capture all of the events and plots associated with this series would be extremely difficult which is why we will only try to capture a few items. The items in the database that the user can change are characters, noble houses, locations, events and a table with events and characters.

Database Outline, in Words

The entities in our database are:

- **Character -** This will be the main entity. It will have a relationship with every other entity in the database. Attributes:
    - **Auto Incrementing ID (int):** Similar to the example database, an id will be auto-assigned to each character when they are recorded in our database. An auto-incrementing number which is the primary key.
    - **fname (varchar):** This will the first name of the character and can be a maximum of 30 characters. It cannot be left blank and there is no default.
    - **lname (varchar):** This will the last name of the character and can be a maximum of 30 characters. Last name can be NULL since some characters do not have last names.
    - **House (int):** This is the house that the specific character is associated with. There are a variety of houses to choose from. Since a character can only be from one house the House ID will be added here. (Foreign Key)
    - **Species (int):** White Walkers, Wights, Giant, Children of the Forest, human, dragon, warlock. Human is the default. (foreign key)

- **Origin (int):** King's Landing, Iron Islands, Dorne..etc. The location_ID corresponding to their origin will be used. (Foreign Key)
- **Weapon (varchar)**: This attribute will be the weapon most associated with the character. For example, Jon Snow and Longclaw. It will be a varchar with a limit of 30 characters. Characters with multiple weapons will only have one weapon listed, preferentially if the weapon is made of valyrian steel.
- **Status (enum):** Is this character alive/dead/wight. This attribute will default to alive.
- **Organization(varchar):** Some characters are involved in certain organizations in the Game of Thrones universe. This includes organizations such as the Kingsguard and Nightswatch. It will be a varchar with a limit of 60 characters.

- **Noble House -** There are hundreds of noble houses in this universe that a character can be a part of.
  - **Auto Incrementing ID (int):** An id will be auto-assigned to each Noble House when they are recorded in our database. An auto-incrementing number which is the primary key.
  - **Location (int):** This is a separate entity in this database so the location_ID will be inserted here. (Foreign Key)
  - **Head (int):** A character that is in charge of a noble house. This attribute will default to none. Character_ID will be inserted here.
  - **Status (int):**
    - Great House (Stark, Lannister) - There are 9 Great Houses that control a number of vassal/lesser houses.
    - Vassal (Tully, Frey) - classified as a noble house that is controlled by a Great House.
    - Extinct (Baratheon, Martell) - an extinct house no longer has any living descendants. Though a bloodline may still exist in bastard children. (foreign key)
  - **Sigil (varchar) -** brief description of the house sigil

- **Location** - Since we want to reference location in other entities, we decided to make the location a separate entity.
  - **Auto Incrementing Id (int):** An id will be auto-assigned to each location when they are recorded in our database. An auto-incrementing number which is the primary key.

- **Name (varchar)**: There are many locations, each belong in a region and on a continent. It will be a varchar with a limit of 60 characters and it cannot be NULL.
- **Region (varchar)** - Dorne, The Vale, The North, Iron Islands. This is where a majority of the plot takes place. However, there are up to 40 regions in all of the game of thrones universe. It will be a varchar with a limit of 60 characters and it cannot be NULL.
- **Continent (int) :** There are three known continents in the Game of Thrones universe, Westeros, Essos, and Sothoryos. (foreign key)

- **Events -** this entity summarizes the location, name, and characters involved in the major plot points. It can be interesting to query whether a character was involved during the event or where they were located at the time of the event.
  - **Auto Incrementing Id (int):** An id will be auto-assigned to each Major Plot Point when they are recorded in our database. An auto-incrementing number which is the primary key.
  - **Location (int):** This is a separate entity in this database so the location_ID will be inserted here. (Foreign Key)
  - **Name (varchar):** The title of the major plot point. For example, the capture of Tyrion or the red wedding.
  - **Season (int) -** the season number that the event occur in
  - **Episode (int) -** episode number of the event
  - **Summary (varchar) -** a brief paragraph summary of the event

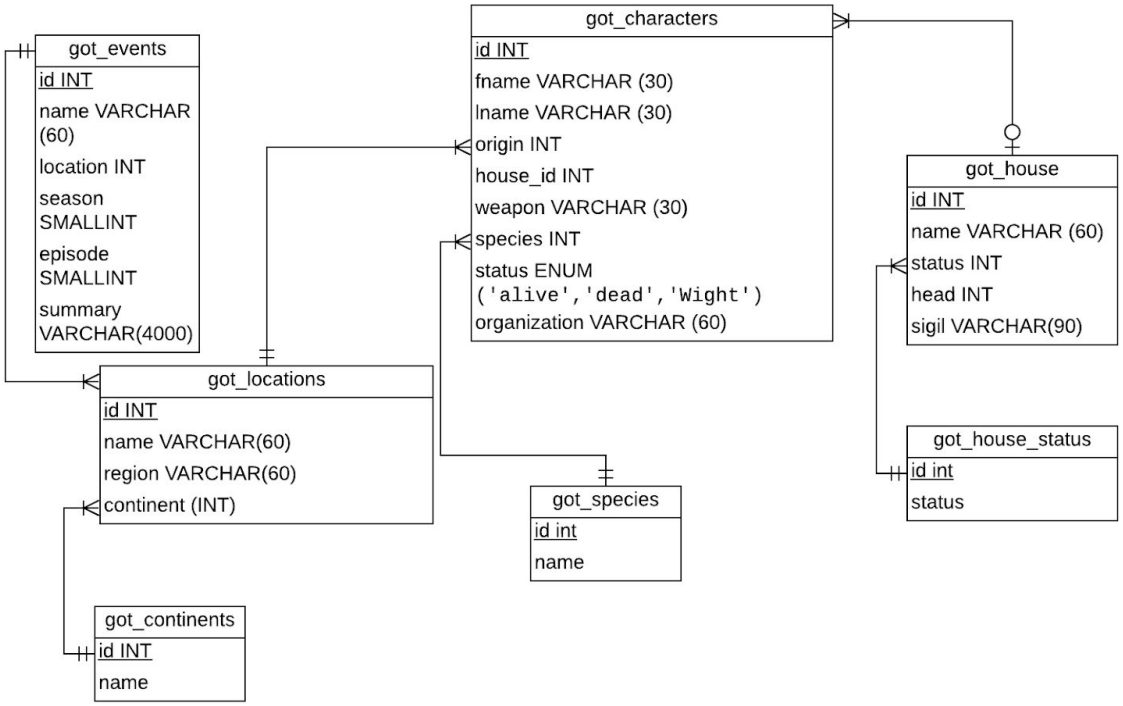
Design decisions

The user will be presented with a table of the entity that they click on. User may create, update, and delete any of the entries. To add a new row, click on the green add button. This will display a form to create a new row. To update a row, click on the pencil icon associated with the row. To delete a row, click on the red garbage button to delete the row.

Project Step 4 Draft Version: CREATE & READ Operations

Danny Tran and Jesse Madrigales

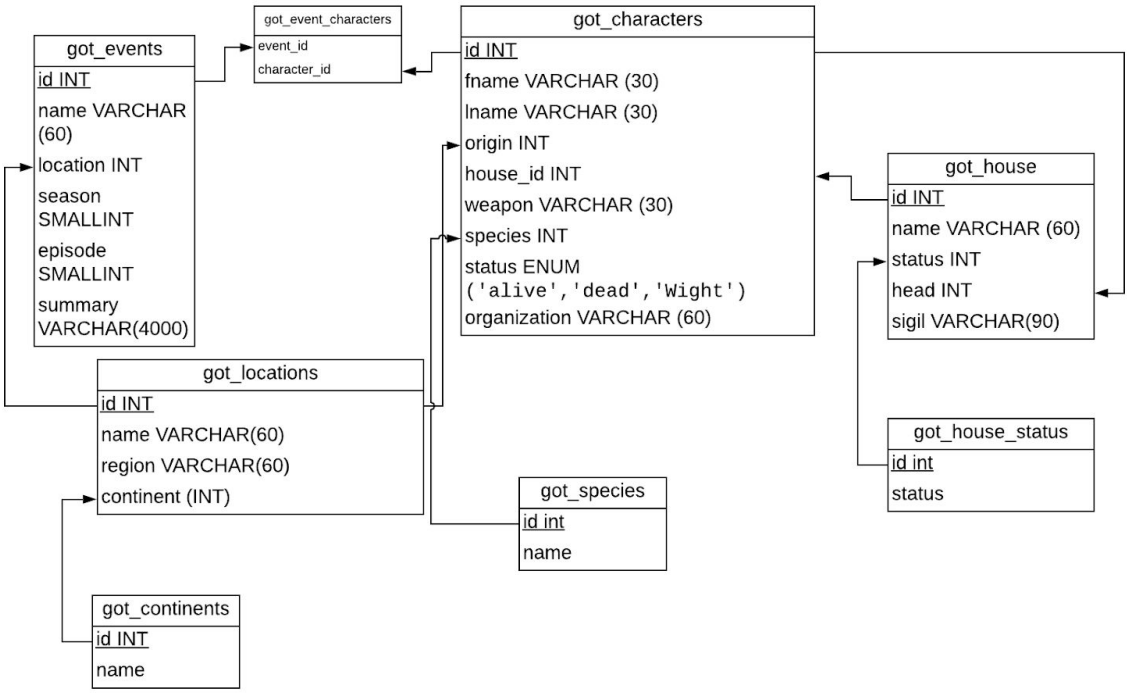
ERD



Project Step 4 Draft Version: CREATE & READ Operations

Danny Tran and Jesse Madrigales

Schema



b) Fixes based on Feedback from Previous Steps:

Feedback fixes from Step 3

- Changed many of the enum attributes into their own tables due to feedback that enums can present many challenges and are considered bad practice. Character status was left as enum to denote health status.
    - Added got_continents, got_house_status, and got_species to replace enum columns.
- Changed a database column called ‘regions’ to allow null and default to null. This fixed an issue where empty strings where being sent through the app despite having a javascript function checking for empty strings.
- Bug when characters got deleted houses where also deleted, changed all foreign key constraints to ON DELETE SET NULL

Feedback from Step 1:

Our feedback consisted of: “Very good!”.

Project Step 4 Draft Version: CREATE & READ Operations

Danny Tran and Jesse Madrigales

However, we did make some changes from the Step 1 submission. The most significant one was removing the alliance entity and replacing it with the location entity. Since we called out location in more than one entity we decided to expand it and make it it's own. We will leave the option to add the alliance entity later in the project. In our original version, we only included entity IDs in the character entity so we added an ID to the other entities. Finally, we added a couple more relationships.

Actions based on the feedback

Character entity

- divided name attribute to fname (varchar) and lname (varchar) per feedback. Fname cannot be null, whereas lname can be null as some characters do not have a last name.
- Marked primary key with an asterisk in the ER diagram.
- Updated ER diagram relationship between Character and Location entities to now be interpreted as "Every character originated from only one location, and every location can be related to one or many characters".
- Added 'weapons' attribute per feedback and our group's discussion of including it.

Noble House entity

- updated 'current head' attribute from varchar to int, which would be a foreign key from the Character entity. Renamed to just 'head' for less confusion with extinct Houses.
- removed 'former head' attribute per feedback to reduce complexities.
- Updated House-Character ER diagram relationship to now be interpreted as "A house has one or many Characters, and a Character can only have at most one House."

Location Entity

- Updated House-Location relationship to be interpreted as "every noble house has 1 location, and every location has zero to many houses."
- Updated Character-Location relationship: "every character is from 1 location and a location has one or many characters".
- Characters with unknown origins will have their origin referenced by a row called "unknown origin" in the Location entity, instead of just leaving this attribute as null.

Major plot points entity

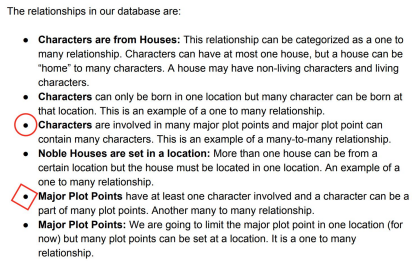
- Updated Major plot points-Location ER diagram relationship to now be interpreted as "A plot point can only have 1 location, and a location can have one or many major plot points".

Upgrades to the Draft version

- Removed redundant relationship found in draft.

Project Step 4 Draft Version: CREATE & READ Operations

Danny Tran and Jesse Madrigales



The relationships in our database are:

- **Characters are from Houses:** This relationship can be categorized as a one to many relationship. Characters can have at most one house, but a house can be "home" to many characters. A house may have non-living characters and living characters.
- **Characters** can only be born in one location but many character can be born at that location. This is an example of a one to many relationship.
- **Characters** are involved in many major plot points and major plot point can contain many characters. This is an example of a many-to-many relationship.
- **Noble Houses are set in a location:** More than one house can be from a certain location but the house must be located in one location. An example of a one to many relationship.
- **Major Plot Points** have at least one character involved and a character can be a part of many plot points. Another many to many relationship.
- **Major Plot Points:** We are going to limit the major plot point in one location (for now) but many plot points can be set at a location. It is a one to many relationship.

- Removed relationship between Noble House and Major Plot Points in ER Diagram. We never mentioned it in the relationship section and it will not be a focus in database queries since there is already a major relationship between location and characters which have a relationship with noble house.

Other changes (Since Step 2 Final was turned in)

- Major Plot Point entity is now called Events. Just a design decision. We felt major plot points was too ambiguous.
- Added warlocks and dragons to 'species' attribute in Characters entity per group's discussion of including them.
- Added 'sigil' attribute in House entity per group's discussion of including it.

All updates to this point should be reflected on the ER diagram and Schema.

-