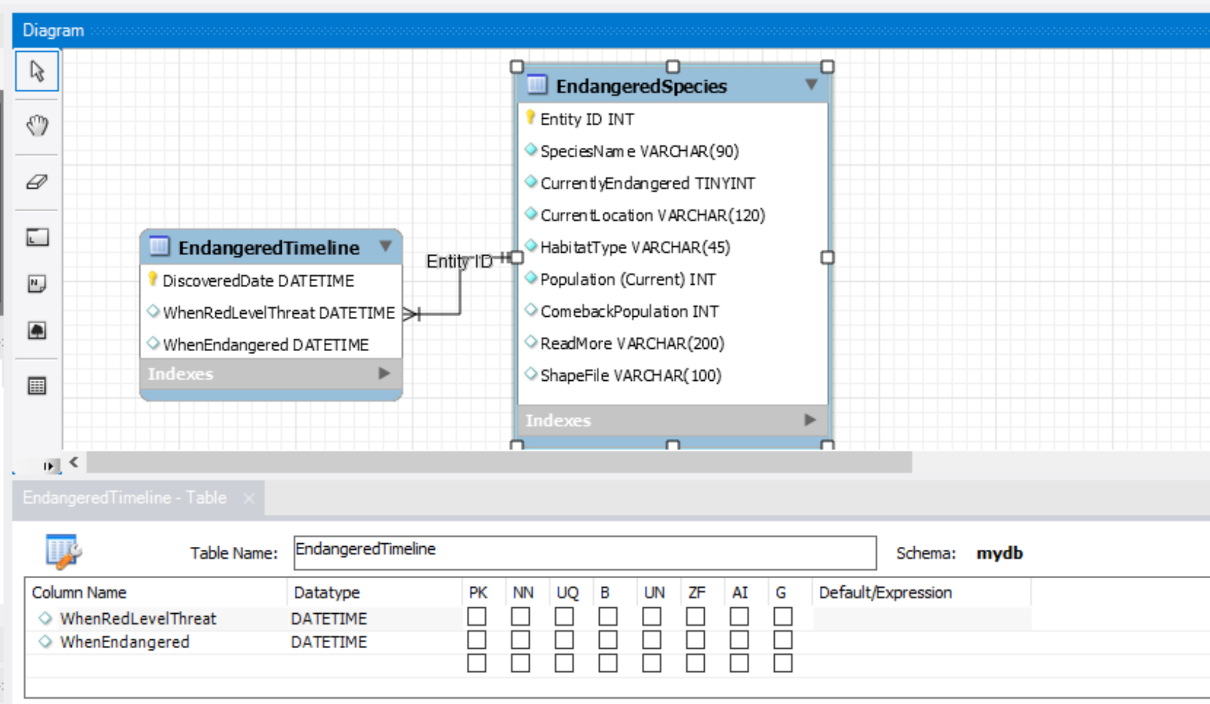
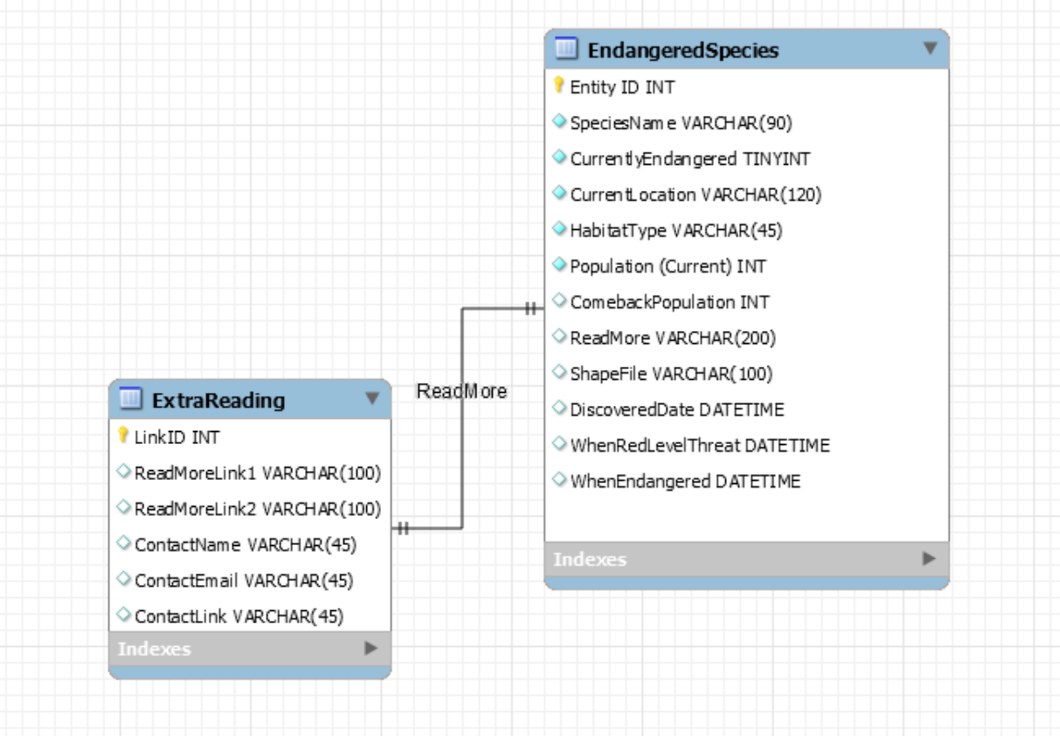
Things Accomplished:

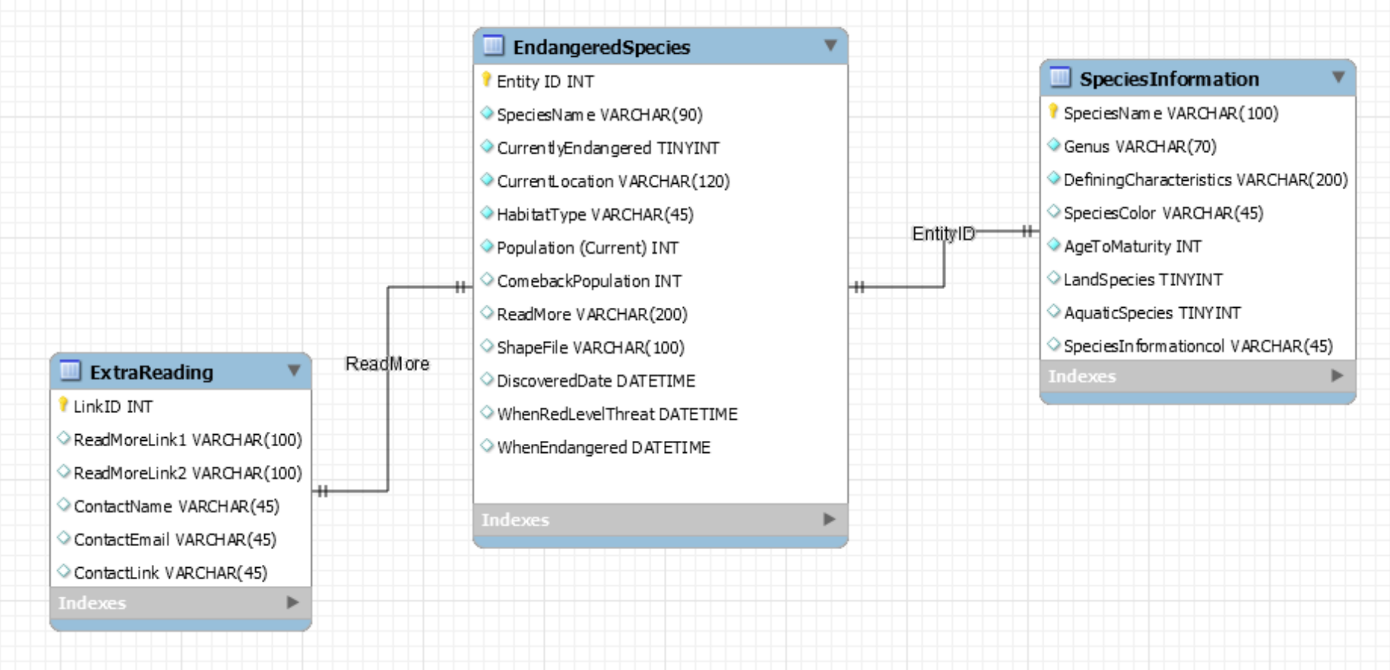
* 1. MYSQL Model is completed, I have documented my process below and shown some of the creative process that goes into making a fully functioning model from a rough sketch.
  2. I started with my Main Table (Endangered Species) to include the main details of each of the species such as name, location, habitat, and other very key characteristics that someone looking at the table may need.
  3. Along with that, I decided to add my timeline table, but realized in the process that this information is better left on the same table.



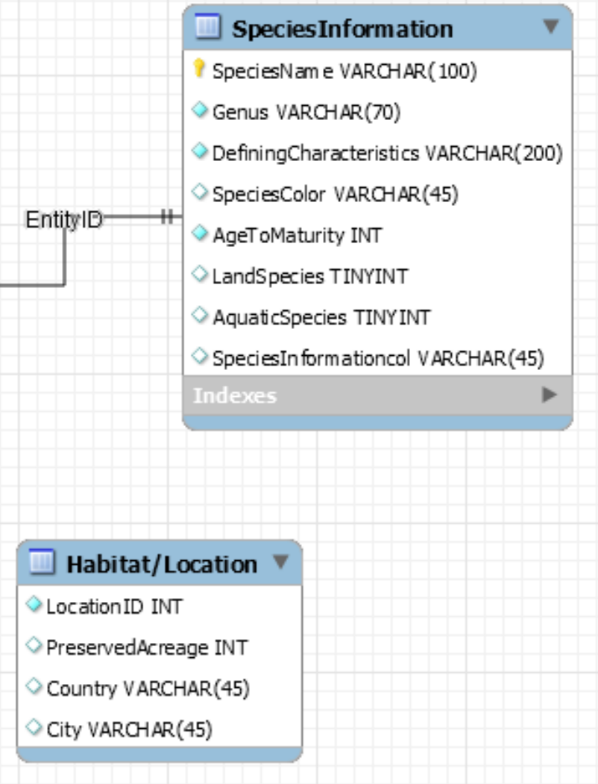
* 1. So I merged My "Endangered Timeline" into the main table, and then created my Extra's table. This table to me makes sense intuitively because this table will hold extra details that only certain users of the table may be interested in. I added a Primary Key called LinkID, and then contact information columns.
  + It will be connected to the main table with a foreign key, but I don’t know whether the foreign key should be EntityID, or a ReadMore Identifier which would also be some unique non-repeating key.



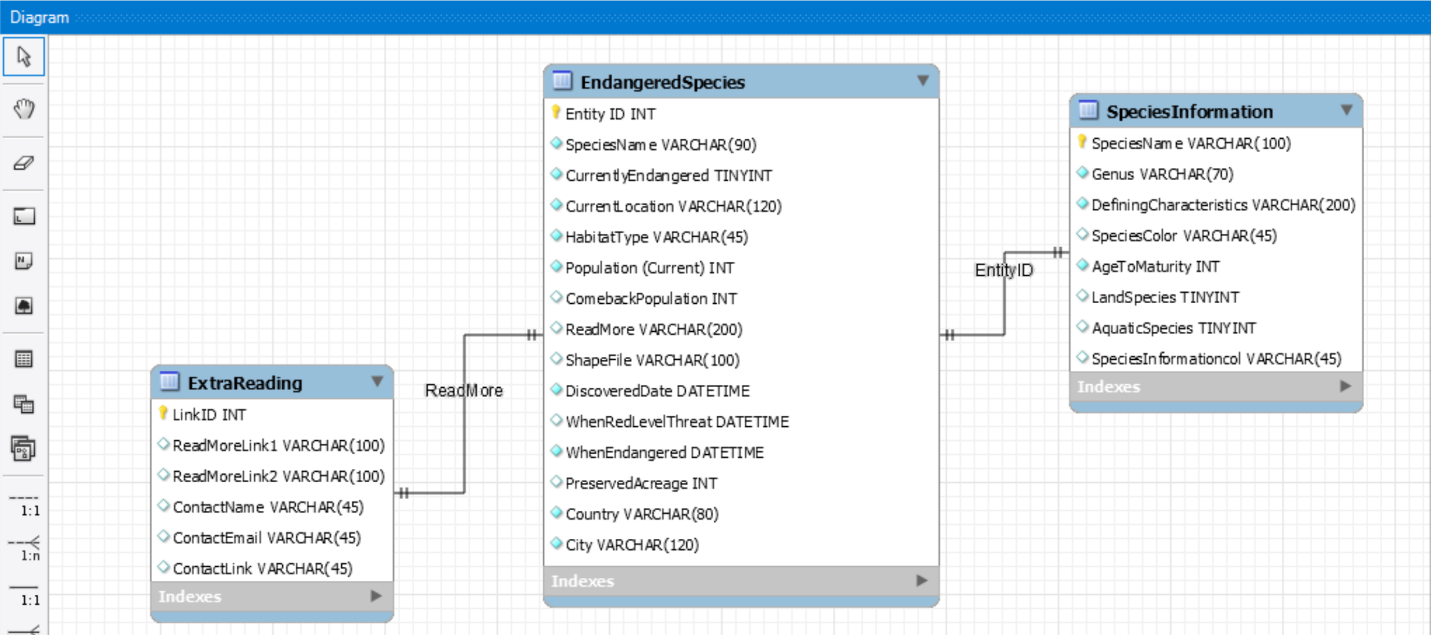
* + Species Information was also broken out into another table. This is also intuitively sound to me because it breaks out extremely specific information about the animal in question. Not everyone may need to see this table, except for those looking for the nuanced total information about the species.



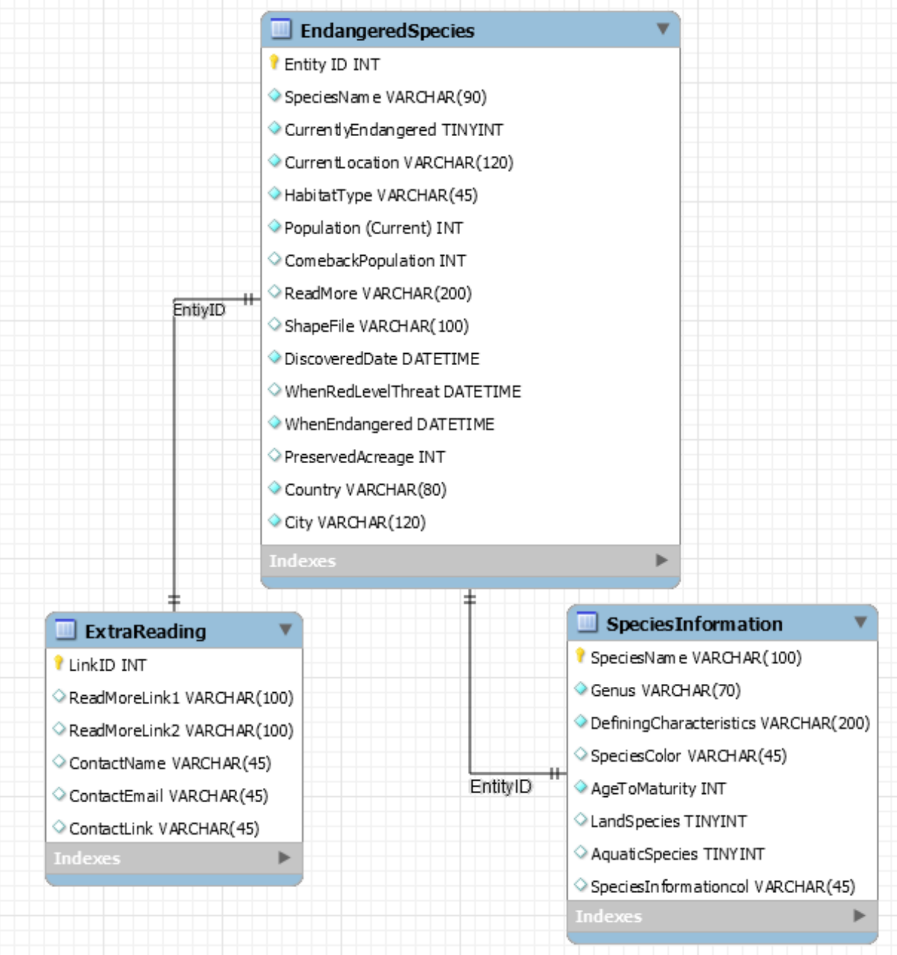
* 1. Now the foreign key connecting those two tables should be the EntityID, with which you will be able to access this Species Information table. This will closely detail everything around the Species.
  2. After this, it was time to add a table about Habitat and Location, another table from my rough draft that I intended to include here but as I thought about it, I realized again that this could all be integrated into the main table.



* 1. I concatenated this table into the main Endangered Species table, and got some final rough sketch of the model now.



* 1. I ended up changing the foreign key to Entity ID on both sides of the table.



* 1. Now that I have my final model, populating it with the correct valuables from the excel sheet will be the tricky part. I have to define some kind of relationship between columns of the table to columns of the Excel sheet.

**ENTITY ID:** (Can I create this manually instead of using the one in the Data? I want SQL to manually increment it with unique, non-repeating numbers).

**SPECIES NAME:** Common Name

**Currently Endangered:** ESA Listing Status (Will need to replace Region \_ with actual Value, because users will have no idea of knowing what Region 2 versus Region 3 will mean)

**CURRENT LOCATION:** FWS Lead Region

**Population:** Not in Main Table --> Find other Data Resource

**COMEBACK POPULATION:** Not in Main Table --> Find other Data Resource

**READ MORE:** Will need to be manually set to some unique number.

**Shape File:** Shapefile (A download link will be included)

**HABITAT TYPE:** Critical Habitat Type

**Date Discovered:** Not in Main Table --> Find other Data Resource

**WHEN RED LEVEL:** Not in Main Table --> Find other Data Resource

**WHEN Endangered:** Not in Main Table --> Find other Data Resource

**Preserved Acreage:** Acres and Miles

**Country:** Not in Main Table --> Find other Data Resource (Perhaps within Regional Data)

**CITY:** Not in Main Table --> Find other Data Resource (Perhaps within Regional Data) --> Where Listed could also be a potential way to summarize both these columns.

**SPECIES NAME:** Common Name

**Genus:** Scientific Name

**Defining characteristics:** Not in Main Table --> Find other Data Resource

**Species Color:** Not in Main Table --> Find other Data Resource (May need to remove this altogether)

**Age to Maturity:** Not in Main Table --> Find other Data Resource (May need to remove)

**Land Species:** Species Group

**Aquatic species:** Species Group

**Species Information Col:** Not in Main Table --> Find other Data Resource (Needed?)

**Extra Reading:** Not in Main Table --> Find other Data Resource (Provide links) (All optional)

* 1. Now I realized at this stage that there are a lot of things that actually need removing…
  + I will highlight the column names that I think are optional or don’t need?

Legend:

Need to be collapsed into one column

Need to be taken out of table entirely

Optional

**ENTITY ID:**

**SPECIES NAME:**

**Currently Endangered:**

**CURRENT LOCATION:**

**Population:**

**COMEBACK POPULATION:**

**READ MORE:**

**Shape File:**

**Date Discovered:**

**HABITAT TYPE:**

**WHEN RED LEVEL:**

**WHEN Endangered:**

**Preserved Acreage:**

**Country:**

**CITY:**

**SPECIES NAME:**

**Genus:**

**Defining characteristics:**

**Species Color:**

**Age to Maturity:**

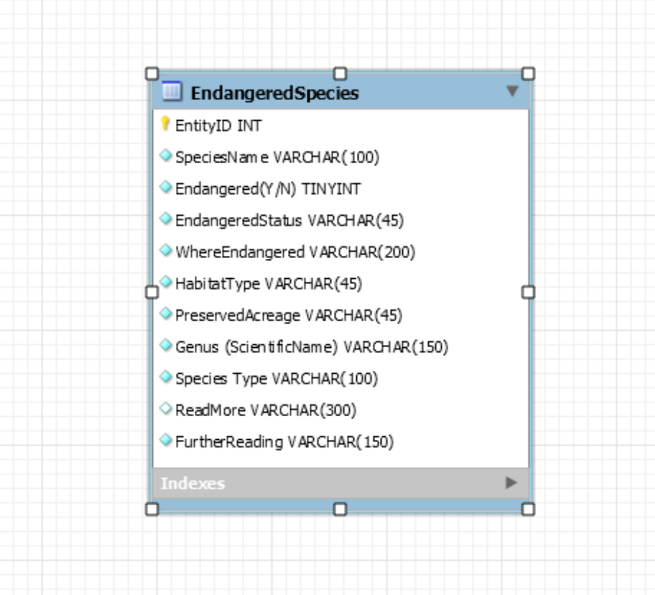
**Land Species:**

**Aquatic species:**

**Species Information Col:**

**Extra Reading:**

* + After this, I checked my other datasets, and could not find much more information in them that the first table didn't include, leading me to believe that many of this table would need to be further adjusted and fit the data contents that I have available with me. Sadly, I don't have much other largescale data to work with that describe more about the species. It'd need to be a purely manual process.



* 1. In the end, it all condenses down to this. One little table!

**ENTITY ID:** Entity ID (Can I create this manually instead of using the one in the Data? I want SQL to manually increment it with unique, non-repeating numbers).

**SPECIES NAME:** Common Name

**GENUS:** Scientific Name

**Endangered (y/n):** Need to write a script where it sees what the Endangered Status is.

**ENDANGERED STATUS:** ESA Listing Status

**Where Endangered:** Where Listed

**READ MORE:** Federal Register Publication

**Shape File:** Shapefile (A download link will be included)

**HABITAT TYPE:** Critical Habitat Type

**Preserved Acreage:** Acres and Miles

**Further Reading:** Script that fills that column with a link to ECOS site.

* 1. Need to look into Lead Regions, and then populate table!