GPS Collar Database

By Alex McComb

GEOG 6150-002

December 4, 2017

Introduction

The database that has been created addresses the increased use and deployment of GPS collars on various species at the Department of Wildlife Resources (DWR). The DWR has been tracking animals with the use of GPS collars for many years now, however, recently we have increased our deployment to better understand animals migrations to aid in conservation and planning efforts. The data being sent from the collars occurs at regular intervals ranging from every few hours to twice a day. With hundreds of collars deployed and being deployed, a functioning and well thought out database is essential so that the data are accessible and orderly for users. The function of the database is to allows us to query very specific data, create maps and graphs, and to make important decisions based on our findings.

2. Integrity Constraints

The integrity constraints, or business rules, of the database have many different requirements. The main requirement is a unique ID for every animal that is, or has been, collared. The collar serial number will not work for this, as the collar can be redeployed to a different animal, or an animal may receive a new collar, thus rendering the collar serial ID non-unique. The unique animal ID will be required to relate the locational and aspatial data with each individual. Using the unique animal ID as the foreign key in these tables will allow us to query specific animals and find information about them. For instance, if we wanted to map the movement of a deer, we would be able to create a definition query using the unique ID in the point data to visualize the coordinates that have been recorded. The point data will be based on a 1 to many relationship between the unique animal ID because there will be many points for one unique animal. The unique ID will also be made up of information about that particular animal, being a combination of both numbers and letters, such as the year it was deployed, followed by the species type, and finally a sequential number starting at zero. This will help identify the species and give a quick overview of the animal while being viewed in the database. The point and line data will be required to be connected to the unique collar ID, as these are the most important data that are required for the database.

3. Conceptual Model Description

The CollarID, or the unique animal ID, will be at the center of the diagram as this is the most important part of the database that allows us to define individual animals and gather information about them. The serial data and the collar data will be made up of a many to many relationship. As stated before, this is because an individual animal can have many different collars, thus many different serial numbers, and a serial number may be put on many different individuals in the case of redeployment of a collar. Due to the large amount of data, loading it in software can be time consuming. To address this, the species will be grouped into different tables so we can query information by species only, and will be a 1 to 1 relationship with the unique animal ID as it is represented the same in this table. A query using this data would look something like “find all Elk over the age of 3 that is female”. Next, the point data will be connected by a line dataset that is in order of date to show the movement of the animal and the distance between each point. These data are one of the most important parts of the database because the data will be heavily focused on mapping. The line data will be a 1 to 1 relationship with the unique animal ID as entire line segment represents only one individual, and this will not be split or shared with other unique animals. There will also be a table that has health data about the animal upon receiving the collar, and this will also be related through the unique animal ID. This data will contain information if a sample was taken, and also information about the overall health of the animal such as teeth and fur health, or if the animal is injured. This will be a 1 to 1 relationship with the unique animal ID because not every animal collared will have health information recorded. None of this data are required, meaning there is no total participation here. Finally, a table representing if the collar is active or not will be connected to the unique ID, and is a 1 to 1 relationship as not all animals are active. This will allow us to query only active animals if needed.

Ancillary data will include elevation, landcover type, and a road dataset that will not be directly connected to the other data, but will be used to tell us more about the data. For instance, we can determine average elevation of a species during a certain time of year, and see how much that changes based on the seasons. Landcover type can tell us if the animals are congregating in specific landcover types to help us understand the preferences of the species. Finally, the road dataset can show us where underpasses for a species may need to be placed based on frequent crossings in a certain part of road, or where fences may need to be put up to reduce vehicle collisions.

4. Relational Model Description

The conceptual model will then be translated into the relational model. This does not vary much from the conceptual model, however in the relational model we see that the "CollarID" (the unique animal ID) is a foreign key in the other data tables. We also show that the many to many relationship between the serial collar number and the unique animal ID is connected by a relationship class containing both primary keys, and allows us to query data between these two tables. This was done using a link table. We also see the data types which are either integer or string based. Here we can see the CollarID is a string type, as it will be made up of numbers and letters, while other primary keys will generally be integers, as they do not require added information to them.

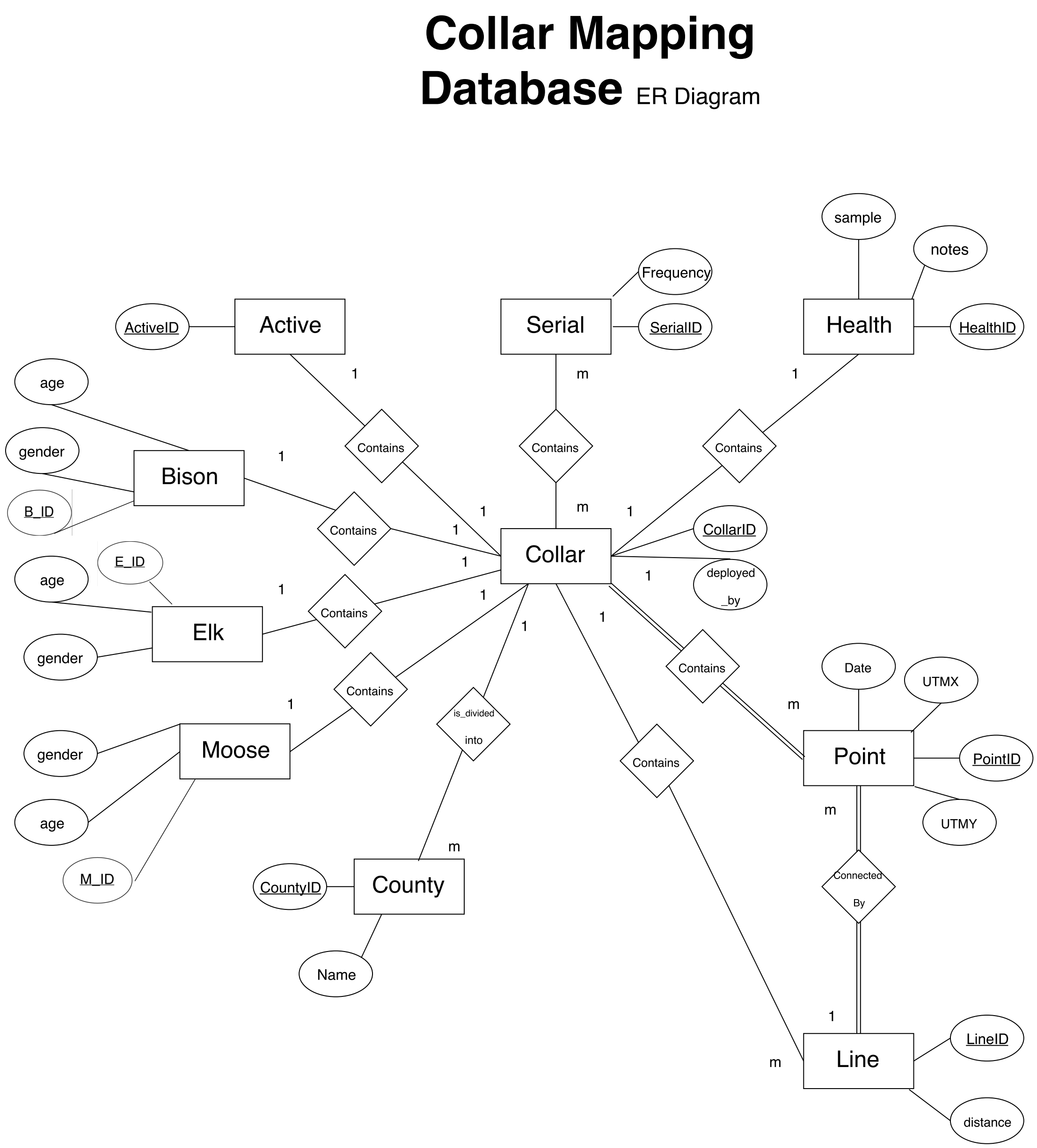
5. Physical Model Description

The physical model is very similar to the relational and conceptual model, however many different relationship classes were created to query data based on the unique animal ID. These relationship classes were created between all the data and the the collar table. As an example, a function that will be beneficial based on a relationship class could be using the relationship between the Active table and the Collar table. Specific collars can be selected that are active to find out who they were deployed by in the Collar table. This will be useful if we need more information about a deployment data so we can contact whoever deployed the collar. This is done by clicking on the relates button at the top of the attributes table and choosing the related table. This can be done with all of the other data, such as selecting a subset of the collar data, and relating it to any other table to find the information about those specific individuals, such as their health data, their serial numbers, or their point and line locations. Another example would be wanting to visually see all points that were female Elk. First, I would join the points table with the elk table. Then I can use a definition query to show only elk that have the attribute female. As stated before, this is efficient in querying specific data since there is so much data to work with and can be very slow loading in software.

6. Future Directions

An important future direction this could take is incorporating this database into a web format, so that people without GIS skills can still utilize the data. This will simplify the database and show only the most important data that can be queried by other novice users. Moreover, a web map can be created to inform the public about certain areas of concern, such as where there are frequent deer crossings to warn them, or give them information about what is affecting animal migrations. Other than that, I feel that the database is fairly robust and doesn't require any other future directions.

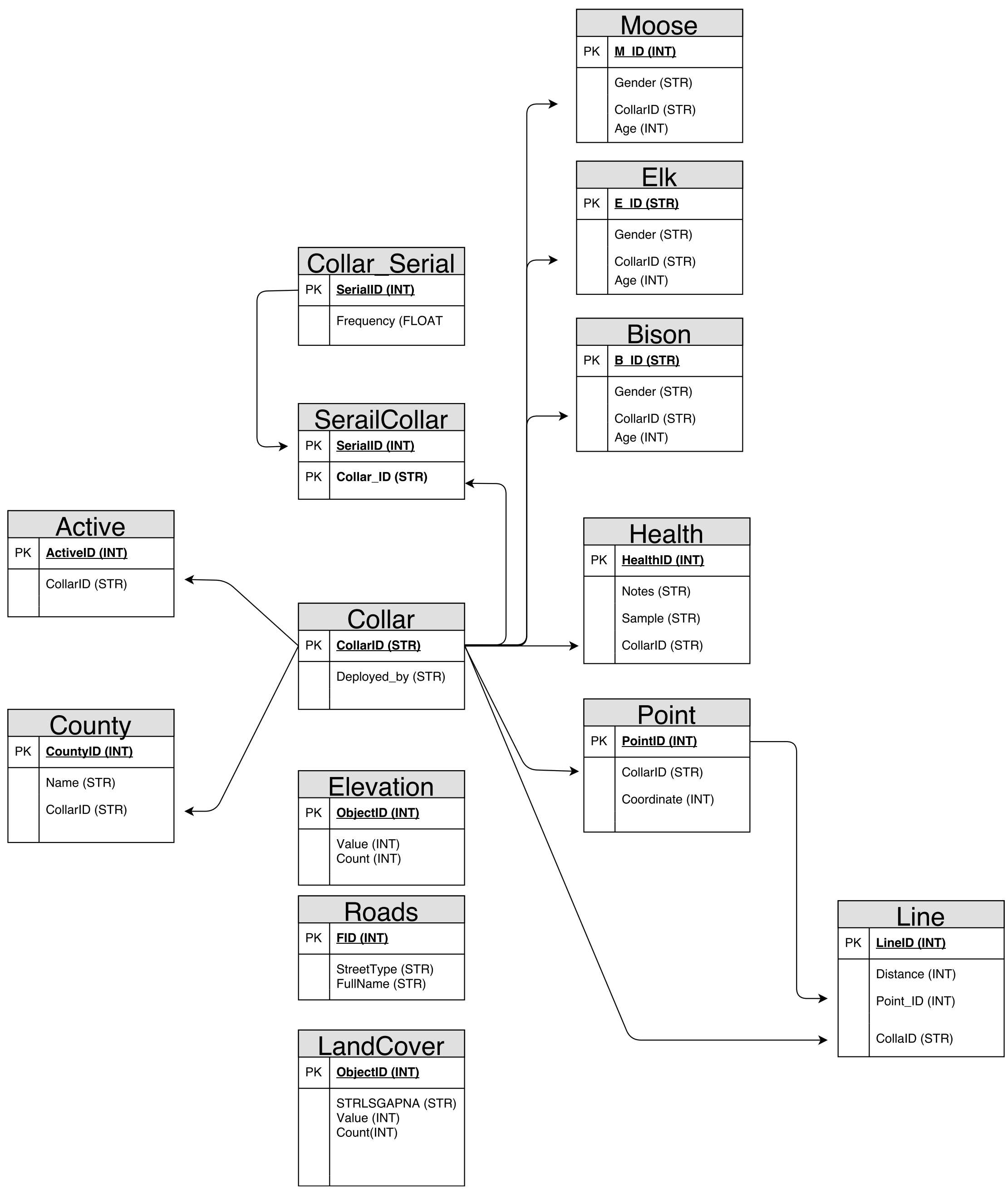
Conceptual Model / ER Diagram



Logical Model / Relational Diagram

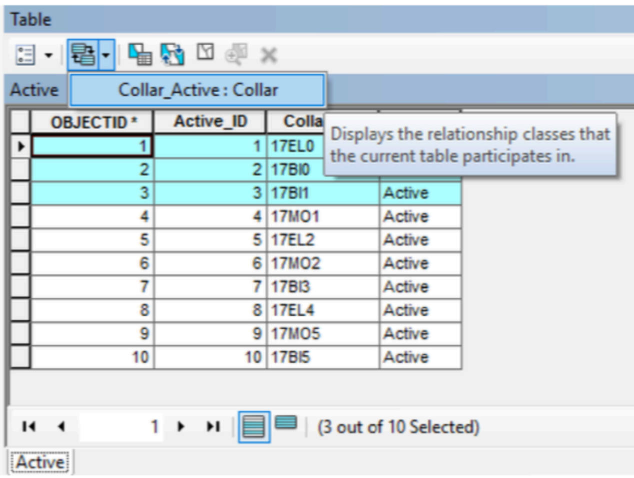
Collar Database

Relational Model Diagram

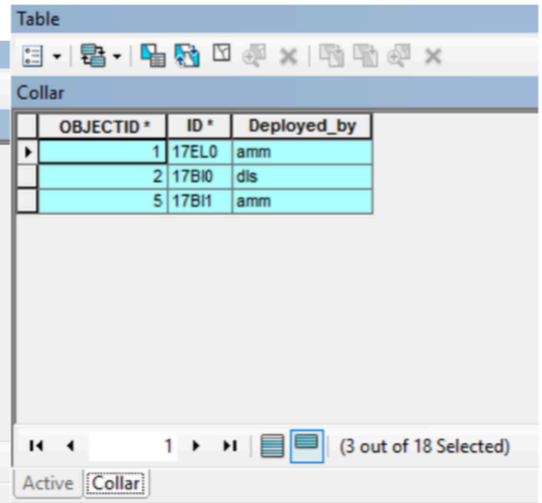


Physical Model

1. The first query, or function, is using the relationship between the Active table and the Collar table. I selected collars that were active to find out who they were deployed by in the Collar table. This is done by clicking on the relates button at the top of the attributes table and choosing the related table.

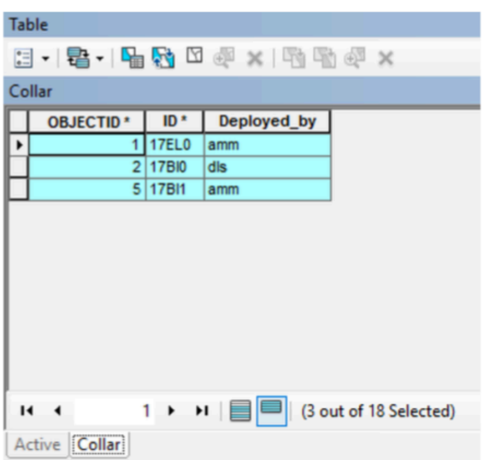


OBJECTID *	Active_ID	Colla	
1	1	17EL0	
2	2	17BI0	
3	3	17BI1	Active
4	4	17MO1	Active
5	5	17EL2	Active
6	6	17MO2	Active
7	7	17BI3	Active
8	8	17EL4	Active
9	9	17MO5	Active
10	10	17BI5	Active

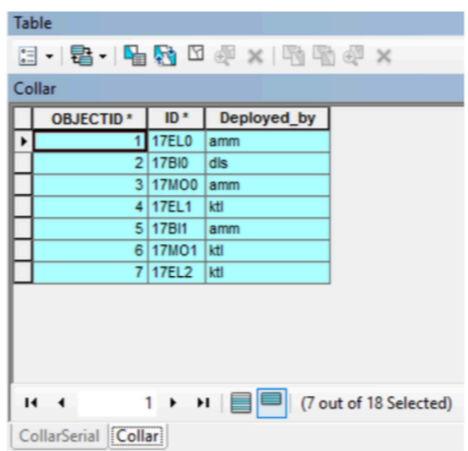


OBJECTID *	ID *	Deployed_by
1	17EL0	amm
2	17BI0	dls
5	17BI1	amm

2. I wanted to find the AnimalIDs of collars with a certain frequency, so as the same as above I selected to frequencies I wanted the AnimalIDs for, then I clicked the relates table which opened the collar table with the selected records.



OBJECTID *	ID *	Deployed_by
1	17EL0	amm
2	17BI0	dls
5	17BI1	amm



OBJECTID *	ID *	Deployed_by
1	17EL0	amm
2	17BI0	dls
3	17MO0	amm
4	17EL1	ktl
5	17BI1	amm
6	17MO1	ktl
7	17EL2	ktl

3. For the final function and query, I wanted to visually see all points that were female Elk. First, I joined the points table with the elk table. Then I used a definition query to show only elk that have the attribute female. This gave me the results shown on the map below, where the first map is not queried, whereas the second one has.