Project Proposal Phase III

Bryan Wieschenberg, Praneel Pothukanuri, Tra-mi Cao

Database Design

Structural Modifications

Our initial design underwent several changes which affect the structure and overall implementation of our database. We continue to use an EER diagram, however with different subclasses: Every Goat(Superclass) **IS A** Dam(Subclass) or Sire(Subclass). The previous subclass “Kid” was ambiguous and held redundant attributes which could be better specified in different entity relationships. Therefore, our additional implementations include two relationships, “HAS” and “IS_DIAGNOSED”. These relationships will bridge “Goat” with “Vaccine” and “Goat” with “Conditions". This widened our database to keep record of diseases Goats carry which could directly influence abnormal birth weights. Furthermore, vaccination records will be beneficial for our research topic which uncovers vaccination correlation with birth weight.

Implementation Modifications

In addition to the multitude of attributes composing “Goat”, “Dam” not only inherits these superclass attributes, but also contains multiple composite and multi-valued attributes. This entity will therefore contribute to a mass portion of the database size. “Triplet_Avg_wt”, “Twin_Avg_wt”, and “Single_Avg_wt” are essential attributes as they contribute to several derived attributes of “Dam”. For instance, based on respective numbers of the aforementioned attributes, a Dam’s average kid weight can be calculated. We also consider the “Birthrate_Success” of a dam, as this will provide meaningful information to a Dam’s ability to reproduce effectively and at the utmost quality. Quality relates to “Parental_Skills” which is an attribute with domain “Good”, “Fair”, and “Bad”. This will be used for optimized matchmaking between Dam and Sire, which could correlate to positive birth weights and higher rates of reproduction. These implementations will increase the effectiveness and specificity of our database to retrieve dams that fall within a user’s desired attributes.

Database Size

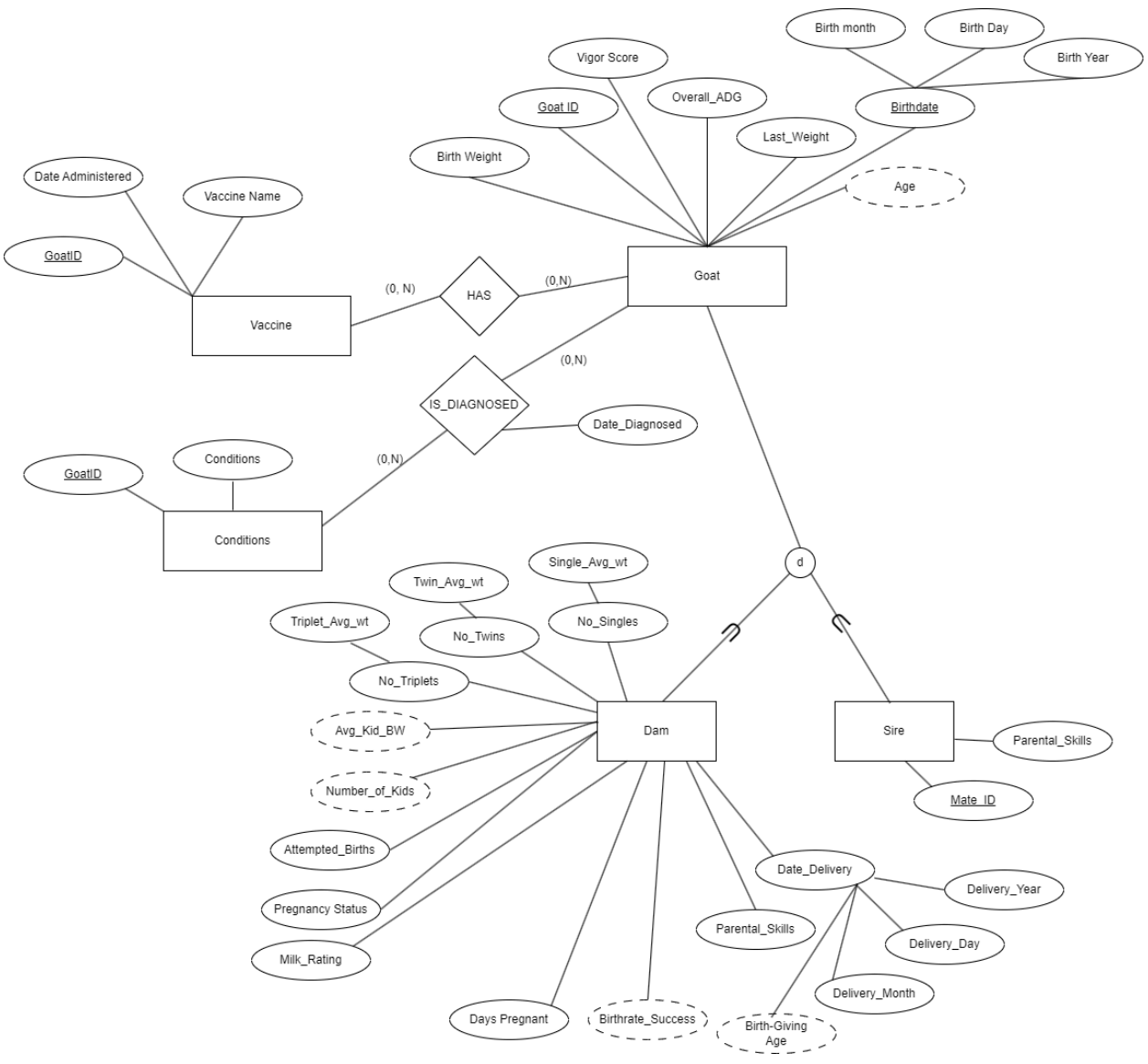
Our database attributes will consist of “char”, “int”, “string”, and “float” values. The size of the database with rough estimation and accounting for null values will fall between a range of 7000-8000 indexes, since there are roughly that many indexes in the Animal.csv if we take into account missing/null values. We believe this number could be dynamic, as goats will constantly be added or deleted in the database. Average search count in a typical work day would consist of various searches and filters per user, therefore, this amount could vary from very low to very high depending on the needs of the farm administrators. We believe this number could range between the values 200-400. Additionally, the estimated number of searches required per query would most likely be 2-3, since the users would only need to apply a few filters to get more specific searches according to our database model.

Use Cases

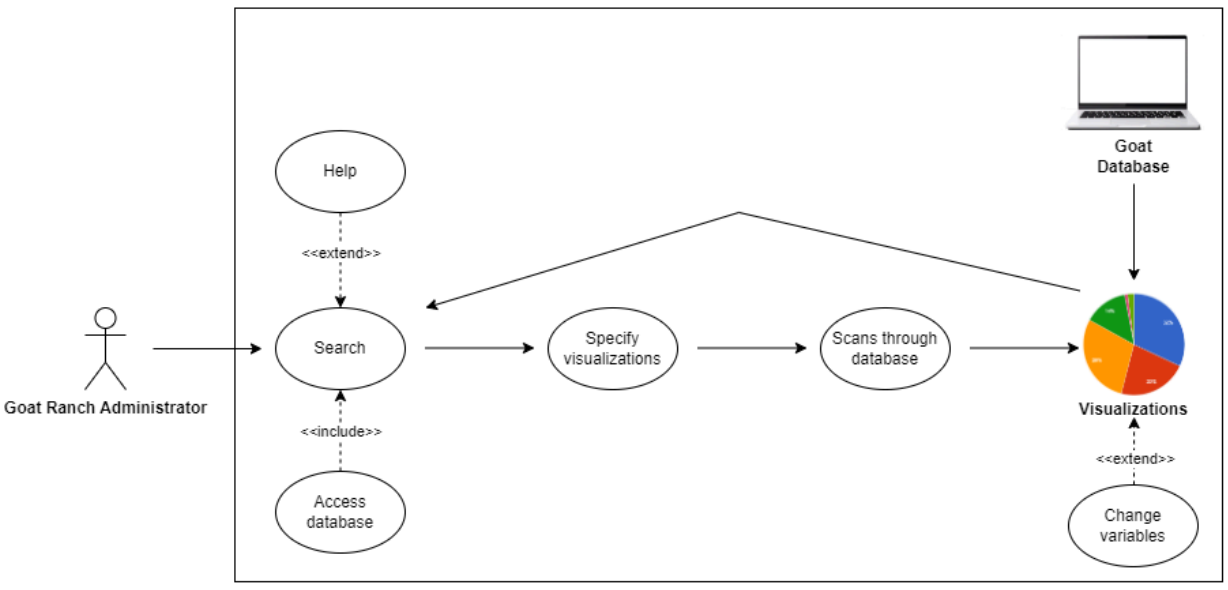
Use Case:	Data visualization - Be able to display graphs of weight, age, and vaccination status. Users can have trends over time visualized and compare different groups of goats based on characteristics

Actors:	Goat ranch administrator
Flow of events:	1. Ranch admin can search through the database to find what they want to be visualized. 2. The user can then select more specific visualizations that the program is capable of displaying. 3. The program will then scan through the database to come up with the appropriate visual, which could either be a pie chart, bar graph, or histogram. 4. The program will then display the visual.
Extension:	1. The program can help the user by showing possible searches they can perform that the program allows. 2. The user can change variables upon having a visual displayed to change the visual in real-time.
Includes:	1. The goat ranch admin can access the database through their device.

Use Case:	Dam ranking - Be able to textually rank goat quality based on a variety of factors, such as weight, age, vaccination status, conditions, and children birthed. Users can filter through certain conditions to have detailed rankings.
Actors:	Goat ranch owner
Success Scenario:	1. Ranch admin can filter through the database to find what types of goats they want to be ranked. 2. The program will then scan through the database to come up with the appropriate textual rankings. 3. The program will then calculate the rankings in an orderly format. 4. The program will then display the textual rankings.
Extensions:	1. The program can help the user by showing possible filters they can perform that the program allows.
Includes:	1. The goat ranch admin can access the database through their device.



Data Visualization



Dam Ranking