Alex Childers
CS 325 - Fall 2018
Last modified: December 7, 2018

How can this implemented database now be used?

I envision this database being used primarily by the *employees* at Loving Care Pet Boarding (LCPB). I think that, using an application interface, employees could input all new information about owners and their pets at once, before enrolling them in any services. This application would percolate all this information about pets and owners down into their respective child tables, which result from multi-valued attributes. More specifically, all information about the owner would be given to LCPB at once, via an online application form or a paper form. Then, the application could run a script to insert the appropriate information into the tables Owner, Owner_phone_num, and Owner_email_addr, using my sequences to generate primary keys and automatically adding foreign keys where appropriate to create relationships. Additionally, all information about pets would be given at once, and the application could percolate the information through the Pet, Pet_vaccine_received, Pet_medication_needed, Pet_diet_restriction, Cat, and Dog tables. The same goes for other entity classes in this scenario -- having it automated would eliminate some of the human error that I came across.

Then, this database can be queried from the employees' end to determine information needed to keep the business running and keep pets and owners happy. As is, a user who knows SQL can query the database using the relationships among entity classes as shown in my model to extract information. For instance, using table joins (which take advantage of relationships), one could now generate a list of ALL pets in the scenario and their corresponding pet type, owner, diet needs, vaccinations, and medication needs all in one fell swoop. The owner of the business can use the Worker table, updating and deleting rows from it as employees came and left. A class teacher could see rosters for all of their classes using equi-joins connecting class section IDs to their unique employee ID. Overall, the database can now be used to keep track of LCPB's day-to-day business operations and to make sure all pets in the building are accounted for and cared for properly.

This could be made even easier using an application-- one could use a query to pull out all connected information from an entity class's base tables and their children. An employee wouldn't have to manually join each table to get the requested information if the application was already aware of the relationships.