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

How can this implemented database now be maintained?

There are two primary items that need to be maintained in the database: keeping information in the database current, and adjusting the database as needed if the client's needs change throughout the time they use it.

If there was an application interface, or there were some SQL-savvy employees, a regular staff member at Loving Care Pet Boarding (LCPB) could perform the task of keeping information in the database current. Assuming that the structure of the tables doesn't change, an employee would simply enter or delete information into the tables and update foreign key references as needed, either manually or through the application. In particular, as new Owners and Pets registered for services, they would be added to the Owner and Pet tables (plus all associated children as needed). As employees and volunteers come and go, the Employee and Volunteer tables (plus all children) would be edited. The Worker_providing_service table would change as employees' responsibilities changed. New classes would need to be added to the Class table, and classes that aren't being offered anymore would need to be removed, and so on and so forth.

However, there are some issues with that -- updating all child tables whenever you alter the record for an entity instance's base table is tedious and very prone to human error, and it's possible that if the records were updated manually (or even with a poor application), foreign key violations would creep into the tables over time and data could become inconsistent. An application could help mitigate this by automatically performing an "on update/delete cascade" operation for new information that is added to the tables. This would be particularly helpful in cases such as updating Employee information, which potentially involves pushing updates to the Worker, Worker_phone_num, Worker_email_addr, Employee, and Employee_formal_qualifications as well -- that's a lot of tables to remember to update by hand.

With that being said, I think that maintenance of information could be performed by the employees, but a formal database administrator (DBA) would likely be required to make adjustments to the underlying table structure over time, or even the model that dictates the structure. For instance, if another entity were to enter the business scenario over time -- say, the animal shelter across town, or schools enrolling students in apprenticeships at LCPB -- an entirely new base table would need to be created for the new entity(s), as well as creating possible child tables for the sake of relationships or multi-valued attributes, *and* establishing relationships with foreign keys. This would involve create table statements, foreign key insertions, and possible restructuring of existing tables. Maintenance of the database structure is possible, but it would likely require the expertise of someone like a DBA who is more experienced in SQL*Plus to make such significant, wide-spreading changes (unless one of the employees happens to be very SQL-savvy!).