Jennifer Nguyen

CPSC 408-01

Rene German

Spring 2021

Pet Shelter Database

**PROBLEM**

Pet shelters and rescues are ethical ways to help keep the number of strays low and provide safe homes for animals. Through a pet shelter, new pet-owner matches can be made for lower prices than through pet breeders. Older, formerly neglected, and unwanted pets can find a new home through pet shelters.

The pet adoption process can be lengthy and difficult to navigate for both the clients hoping to foster or adopt, and the employees who have to approve the applications. On the client standpoint, after submitting their application, the lack of contact information with the shelter and updates on their profile status may deter any interest in fostering or adopting. On the employee standpoint, the large volume of applications can be overwhelming. It is easy to neglect applications that have been pending for a long time. Employees are faced with another challenge of matching pets with clients who want them.

**ELEMENTS OF SOLUTION**

***Separate Access Levels***

The solution database features two level of access: the client level, and the employee level. The client level allows clients to change their personal information, check their approval status, and obtain the contact information of the employee assigned to them. For simplicity of the database, clients could only be from California, Oregon, or Washington—states with zip codes starting with 9—and pet breeds were excluded in client pet preferences. These capabilities provide the basis of what a client would use the database for. If the database were to be expanded, clients would input their preferred pet breeds and relative pet age. As of now, the database only allows clients to choose between dog and cat. The client approval status is an important feature. When a new client is added to the database, they default to not approval as of the date they entered the database. When an employee updates the client, the date changes automatically to the date the employee updates the approval status. Realistically, some form of notification, either email or phone, would be given to the client. The approval process of the clients were simplified in the database to approved or not approved yet. Typically, clients are screened based on their living condition, household, and personal lifestyles.

The employee level contains more functionalities than the client level. There are still limitations, because the assumption is that the employee is not upper management. Therefore, the employee is unable to update information on the shelters, such as editing a shelter’s address or adding a new shelter. The information of a shelter is unlikely to be frequently changed. To make the database make sense, each employee is limited based on their shelter and client list. Each employee is only able to change the personal information of pets in their shelter. Each employee is only able to update the personal information of clients assigned to them. In this way, employees are more organized and can consistently check the status of their clients to ensure timely screening is given to each client.

The employee level is in charge of creating matches. When the employee first creates a match, they use a client ID. Then, they enter a pet ID. The client information is compared to the pet information. Any client that has been approved may be a part of a match. The client’s preference must match the pet’s type, i.e. dog or cat. The client’s phase must match the pet’s phase, i.e. fostering or adopting. If all comparisons work, the match is processed and added to the appropriate logging table.

***Data Generation***

The most difficult part of the database was setting up and populating the appropriate tables. Once the Client, Pet, Shelter, and Employee tables were created, the connection tables had to be populated. Rollback was used to error check the files for accuracy before being imported into the tables. For simplicity, employees and pets were assigned to random shelters, and clients were assigned to random employees. The Fostering and Adoption tables are logging tables, logs of all the fostering and adoption that had occurred. To populate this data, eligible clients were selected, i.e. clients that were approved to foster or adopt. Clients were matched to pets according to realistic dates: the client approval date had to be after the birth date of the pet. A natural join was used to generate all possible matches. The natural join was analyzed and actual matches drawn from there. Once a match was made, the pet could not be fostered or adopted again, and the client could not be used again. These matches were inserted into the Fostering and Adoption table. In the Fostering table, the fostering dates were set to be 35 days apart. In the Adoption table, the adoption date was set to be 3 days after the approval date of the client. Once a pet had been fostered, the phase of the pet was changed to “adopting.” Once a pet had been adopted, the pet was soft deleted from the database. The Adoption table was populated after the Fostering table, so data does exist of pets that had been fostered and adopted within this database.

***Views***

Database views were created to store the more complicated queries that had multiple joins. A view was made between the Clients and Employees tables that helped in expanding the relationships listed in the ClientToEmployee base table to include all relevant information. Another view was made between the Shelters, Pets, and Employees tables to combine the relationships in the PetToShelter and EmployeeToShelter tables. The information from these queries are expected to be the most used.

***Statistics***

Statistics are important in evaluating the performance of the shelters. To be able to obtain the numbers on the foster and adoption counts, the Fostering and Adoption tables were used. The Fostering table contained the IDs of the client and pet as well as the dates of the fostering. Similarly, the Adoption table contained the IDs of the client and pet and the date of the adoption. Information can be filtered by date. Using joins, the counts could be filtered by shelter.

***Reports***

Since one of the biggest problems with pet shelters is the length of the approval process, reports could be generated for all the pending application and all the approved applications. The reports order the clients based on the date of the pending or approval status. This allows employees to determine the existing demand and quickly process applications that had been on hold for the longest time.

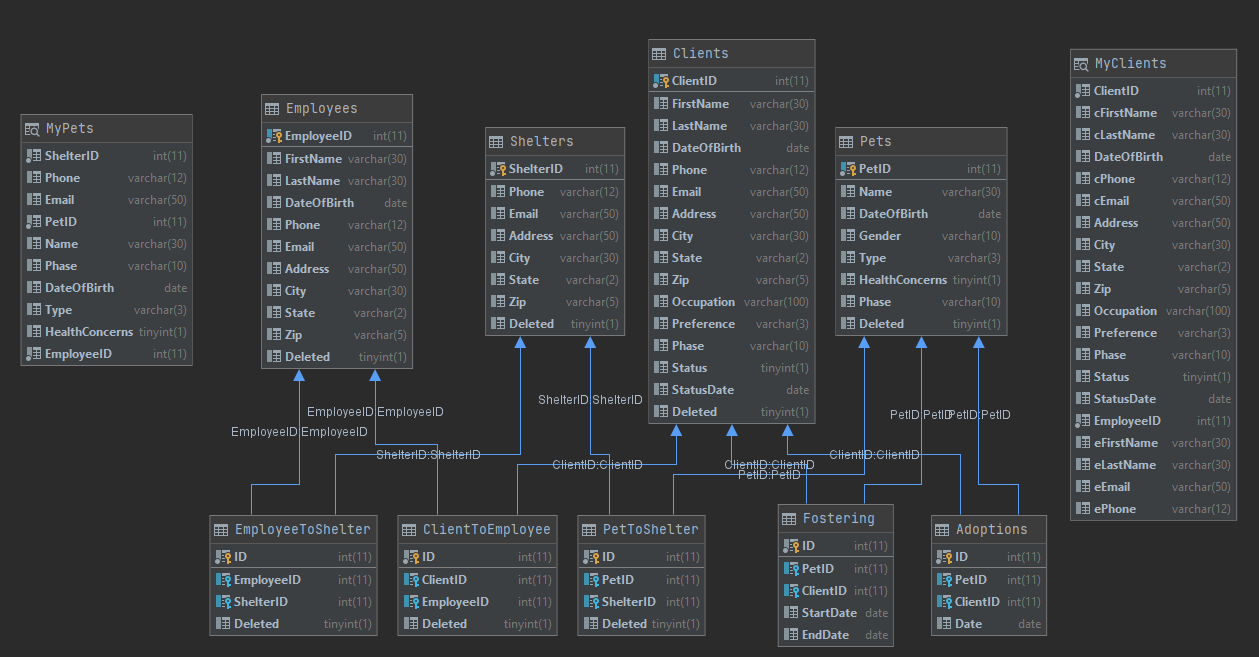
**RESULTS**

The database can be accessed using the command line. Decorative text banners were added to enhance the user experience. Navigating through the database was dependent on the current menu of possible actions. Each action was numbered and users selected a number to proceed. In most menus, the user had the option to exit or return to the main menu. Autocapitalization was used in certain binary options of y/n and for data entry to make the records clean.

**SCHEMA DIAGRAM**

The resulting database consists of 9 tables and 2 views. The Faker library was used to generate data for the Client, Pet, Shelter, and Employee tables. Random matches were made to populate data for the ClientToEmployee, PetToShelter, and EmployeeToShelter tables. The Fostering and Adoption tables, and views were populated as discussed above.

The relationships between the tables are dependent on the assigned IDs. To be clean, foreign keys were named the same as the primary keys. Beside the four information tables, the records in the other tables were identified based on a record ID. Variables that were the same between the Client and Pet tables were Client.Preference = Pet.Type, and Client.Phase = Pet.Phase.

****