Dog Kennel Data
Alexa Williams

CPSC 321, Fall 2024

1 Summary

I have developed a database that holds information relating to dog kennel businesses. The database has many tables holding different key information needed for a dog kennel business to run smoothly. The database holds many types of data saved in organized and interconnected tables to ensure that a dog kennel business can run smoothly. The database is set up in a way that any dog kennel business should be able to add to it as long as they add their own business to the business table because all the other tables and information will connect to the business table in some way. This database simply stores information pertaining to any given dog kennel business.

2 Use Cases

2.1. Adding a new customer/owner to the system -

- Employees need a way to add new customers to the system, so this use case will give them a way to do that. This use case is for creating data.

2.2. Adding a new dog to a given owner -

- Owners often get more dogs that they will also want to have in the system, so this allows employees to add a new dog to the given owner. This use case is for creating data.

2.3. Removing a dog from the system -

- When a dog is deceased, it needs to be removed from the system because a booking can no longer be made for the given dog. This will remove the dog from the system. This use case is for removing data.

2.4. Removing an employee from the system -

- When an employee is fired or quits, they must be removed from the system. This will remove the given employee from the system. This use case is for removing data.

2.5. Updating the Meal Plan for a given dog -

- It is very common for dogs to have changes in their diet. This will allow employees to update the Meal Plan for a given dog when needed. This use case is for updating data.

2.6. Updating an owner’s account balance -

- Employees need to be able to update the account balance of a given customer. This accounts for things like adding a charge for a new booking or clearing the balance after they pay the current balance. This use case is for updating data.

2.7. Employees can search for a customer in the system -

- Employees might need to search for a customer’s account when a customer comes in. They can search by name, and the customer’s that match that name will be listed in order by their id number. This allows an easier way to find things about a customer like their current account balance. This use case is for searching for data.

2.8. Employees can search for dogs in the system -

- Employees might need to search for a certain dog in the system. They can search by name and then the dogs with that name will be sorted on color, owner, breed, and birthday. This use case is for searching for data.

2.9. Find the number of customers that have an old dog in the system -

- The company might want to analyze the number of older dogs that are currently in the system. This will allow the company to find the number of dogs having an age greater than 10. This use case is for analyzing data.

2.10. Find all the reservations that are occurring on a given day at a given company -

- The user can input a date, and the system will return all of the reservations that started on the given date. This allows the company to know exactly which new dogs they are responsible for on any given day. The reservations will be listed in the order of their start dates. This use case is for analyzing data.

2.11. Find all the dogs that have a given number or more reservations in the system –

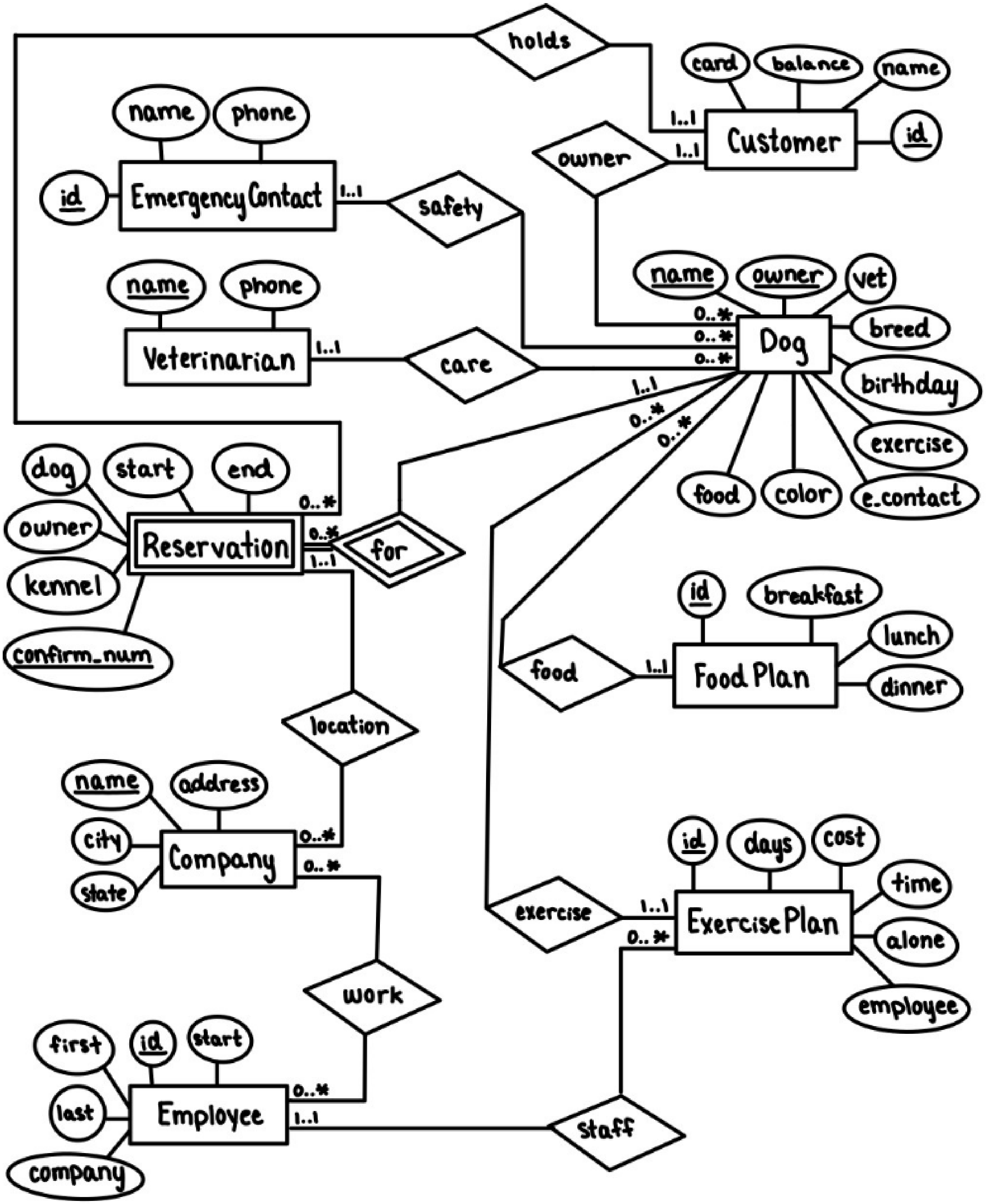
- The user can input a number and the system will return all of the dogs and their owners that have that given number or more reservations in the system. This is just an interesting metric for the company to have. This use case is for analyzing data.

2.12. Find Number of Reservations for each Start Date –

- The system can see how many reservations are starting on any given day. This only shows dates that have a reservation on them, so it can be assumed that any date not listed in the output does not have any reservations starting on that day. This use case is for analyzing data.

3 Logical Design

3.1 Entity-Relationship Diagram



3.2 Relational Schema

customer(id, name, balance int, card)

contact(id, name, phone)

employee(id, first, last, start, card, company)

exercise(id, days, cost, time, alone, employee)
   FOREIGN KEY - (employee) REFERENCES employee(id)

veterinarian(name, phone)

food(id, breakfast, lunch, dinner)

dog(name, owner, vet, breed, birthday, exercise, e_contact, color, food)
   FOREIGN KEY - (owner) REFERENCES customer(id),
   FOREIGN KEY - (e_contact) REFERENCES contact(id),
   FOREIGN KEY - (vet) REFERENCES veterinarian(name),
   FOREIGN KEY - (exercise) REFERENCES exercise(id),
   FOREIGN KEY - (food) REFERENCES food(id)

working(employee, company)
   FOREIGN KEY - (employee) REFERENCES employee(id)
   FOREIGN KEY - (company) REFERENCES company(name)

company(name, address, city, state)

reservation(confirm_num, kennel, dog, owner, start_date, end_date)
   FOREIGN KEY - (kennel) REFERENCES company(name)
   FOREIGN KEY - (dog, owner) REFERENCES dog(name, owner)

4 Use-Case SQL Statements

4.1. List Old Dogs -

This finds all the dogs over the age of ten that are currently in the system.

```
SELECT COUNT(*) AS count FROM dog WHERE birthday <= '2014-01-01'
```

4.2. List Meal Plans -

This simply prints out a list that describes each of the available meal plans.

```
SELECT * FROM food
```

4.3. Add New Customer -

This allows users to insert a new customer into the system by entering values for a new row.

```
INSERT INTO customer VALUES (?,?,?,?)
```

4.4. Add New Dog -

This allows users to insert a new dog into the system by entering values for a new row.

```
INSERT INTO dog VALUES (?,?,?,?,?,?,?,?,?)
```

4.5. Remove Dog -

This allows users to delete any dog from the system.

```
DELETE FROM dog WHERE name = ? AND owner = ?
```

4.6. Remove Employee from Company -

This allows users to remove an employee from working at a certain company.

```
DELETE FROM working WHERE employee = ? AND company = ?
```

4.7. Update Dog's Meal Plan -

This allows users to update the meal plan that a dog is on.

```
UPDATE dog SET food = ? WHERE name = ? AND owner = ?
```

4.8. Update Customer Account Balance -

This allows users to update the current account balance on a certain customer account.

```
UPDATE customer SET balance = ? WHERE id = ?
```

4.9 . Search for Customer -

This allows users to search for a customer’s account in the system.

```
SELECT * FROM customer WHERE name = ? ORDER BY id ASC
```

4.10. Search for Dog -

This allows users to search for a specific dog in the system.

```
SELECT * FROM dog WHERE name = ? ORDER BY color ASC, owner ASC, breed ASC,
birthday ASC
```

4.11. Find Reservations -

This allows users to search for all reservations that occur on any day the user gives.

```
SELECT * FROM reservation WHERE start_date <= ? AND end_date >= ? AND kennel = ?
ORDER BY start_date ASC, end_date ASC
```

4.12. Find Dogs By Number of Reservations –

This allows users to find all the dogs that have a given number of reservations or more in the system.

```
SELECT dog, owner, COUNT(*) AS num FROM reservation GROUP BY dog, owner HAVING COUNT(*) >= ?
```

4.13. Find Number of Reservations for each Start Date –

This allows users to know exactly how many reservations there are starting on each date in the system.

```
SELECT start_date, COUNT(*) AS num FROM reservation GROUP BY start_date ORDER BY start_date
```

5 Applications

I developed a text-based user interface service for this database. The program will be written in Java, and it will use dynamic SQL. PostgreSQL is what I will use to manage this database. This will allow companies to add, delete, and access any necessary data needed.

6 Conclusions

Overall, I have created an application that could be used to assist dog kennel businesses in organizing the day to day information needed for a successful business. If I were to spend more time on this project I would continue to expand the use cases. There are so many different insertions and deletions that could take place in this application, but I simple ran out of time to do a more all encompassing set of them. I would also spend the time to go back and make the database derive a few different columns such as customer id and reservation confirm_num. I learned that I really enjoy working with dynamic SQL during this project, especially when I get to decide what I want to create. I found that it is very fun to explore the endless options that I could make for use cases and dive deep into interesting connections embedded in the data. In the end, if I had more time, I would just continue to expand this application in every way. I would expand the user interface and add more tables on the back end. This would allow me to have an even more presentable product.