

CS 3520 602

Professor Craig Bell

## Fur-ever Homes Adoption Center Database Project

Sofia McNary, Alissia Huntzinger, Bishakha Tamrakar

# Table of Contents

Fur-ever Homes Adoption Center Database Project .....	1
Table of Contents .....	2
Executive Summary .....	2
Vision Objective .....	3
Mission Statement .....	3
Service .....	4
Use Case .....	4
Business Requirements .....	5
ER Diagram .....	7
ERD Description .....	8
Approach Used to Develop Current Solution .....	9
Current releases .....	9
Future releases .....	9
Conclusion .....	10
Appendix .....	10
SQL Table Creation Scripts: .....	10
SQL Data Population Scripts: .....	13

## Executive Summary

Project Title: Fur-ever Homes Adoption Center

The Fur-ever Homes Adoption Center database project is designed to support the operations of a nonprofit, no-kill animal shelter that cares for both cats and dogs. The shelter's mission is to rehome animals by ensuring they receive proper care, medical attention and placement with loving families. Our proposed database system aims to improve how the shelter manages animal intake, adoptions, foster care, volunteer assignments and medical services.

This system tracks a wide range of information, including adopter details, volunteer roles, foster home assignments, breeds and animal medical histories. It enables the shelter to monitor the status of animals housed onsite or placed in foster care, along with their readiness for adoption. Animals not immediately eligible for adoption due to incomplete medical care can be reserved by potential adopters through a built-in reservation feature, helping ensure no animal is overlooked.

From a business perspective, the shelter accommodates up to 200 animals. When this capacity is exceeded, additional animals are temporarily placed with registered foster families. To manage all these moving parts, the database reflects real-world requirements such as limited space, animal eligibility, and staff assignments. It models different types of locations such as foster homes, shelter nurseries, medical bays, and standard rooms with a clear relationship between these spaces and the animals they house.

## Vision Objective

Our vision is to build a clear and easy to use database system that helps the Fur-ever Homes Adoption Center work better and with more care. We want to support the shelter's goal of saving, caring for and finding new homes for cats and dogs by making a system that keeps track of each animal from the time they arrive to when they are adopted. This includes recording their medical care, handling reservations for animals not ready to be adopted and keeping details about foster homes and volunteers.

By making information easy to find and update, our system will help avoid mistakes, save time and make work easier for the people at the shelter. It will also help every animal get the care they need, whether they are at the shelter or in a foster home. In the future, we hope this system will lead to more animals being adopted, better health care for them, and a more caring and organized place while they wait for their forever homes.

## Mission Statement

Our mission is to design and build a database system that helps the Fur-ever Homes Adoption Center keep better track of animals, adopters, volunteers and foster families. We want to make it easier for the shelter to manage animal care, organize adoption and foster processes, and record important details like medical needs and locations. By creating this system, we aim to help the

shelter save time, reduce confusion and give each cat and dog the best chance to find a safe and loving home.

## Service

At our Adoption Center, our main goal is to offer caring and helpful services to both animals and people who want to adopt. We believe every animal deserves a second chance, so we work hard to make sure each pet is healthy, safe and ready to find a loving home. Our center has medical rooms where animals get the care they need, and we also have a fostering program where animals can stay in a temporary home until they are adopted. Our staff and volunteers spend time with the animals to help them feel safe and happy.

We want to make the adoption process easy and special for everyone. When someone wants to adopt, our team helps by answering questions and finding the best pet that fits their life or even their breed. Besides adoption and fostering, we also offer temporary reserve for animals when they need medical services. Our volunteers are very important and help with taking care of animals.

Most of all, our service comes from kindness and caring. We want to be more than just a shelter, we want to be a place where people learn, connect and help animals get a new start. Whether someone chooses to adopt, foster or volunteer every action helps us grow our work. By giving honest, friendly and caring help including medical care and fostering we are making a better life for animals and the people who love them.

## Use Case

Use case 1:

System: pet database

Actor: An individual looking for an a German shephard to adopt

Scenario: The shelter manager can search through our application to see the availability of the animals in the shelters and decide on their match. When they decide if the animal is available, then they can pick up the animal if it is not, then they can reserve the animal until they are ready to be adopted.

Use case 2:

System: pet database

Actor: A manager that needs to check room assignment

Scenario: The manager logs into the system to view which animals are assigned to which rooms and use this information to decide where to place new animals or move animals needing medical care. The system helps them manage space and avoid overcrowding.

Use case 3:

System: pet database

Actor: A volunteer helping with animal care

Scenario: The manager logs into the system to check which animals require care, feeding, walks or medical attention. Based on this information, the manager creates and prints a task list for volunteers.

## Business Requirements

Scope: Creating a database for an animal shelter.

Functional Requirements: Within the database queries should return information about pet availability, breed, age, and location. They should also retrieve info for employee use like medical history.

Non-Functional Requirements: Readability, clear project structure

Stakeholder expectations: The main stakeholders for the Fur-ever Homes Adoption Center database include shelter managers, employees, volunteers, adopters and foster families. These groups expect the system to be easy to use, reliable and helpful in organizing shelter operations. Shelter managers expect the database to help them quickly check animal availability, manage room assignments and keep track of medical and adoption records without confusion. Volunteers and employees rely on the system to know which animals need care, feeding, or medical attention and they expect clear instructions based on up-to-date information.

Adopters and foster families, though not directly using the system, expect their experience with the shelter to be smooth and well-organized. They rely on the staff to have accurate information when helping them choose or care for a pet. All stakeholders expect the system to reduce mistakes, save time and support the shelter's mission of giving animals the best care possible. Meeting these expectations helps build trust, improves daily work and ensures more animals are placed in safe and loving homes.

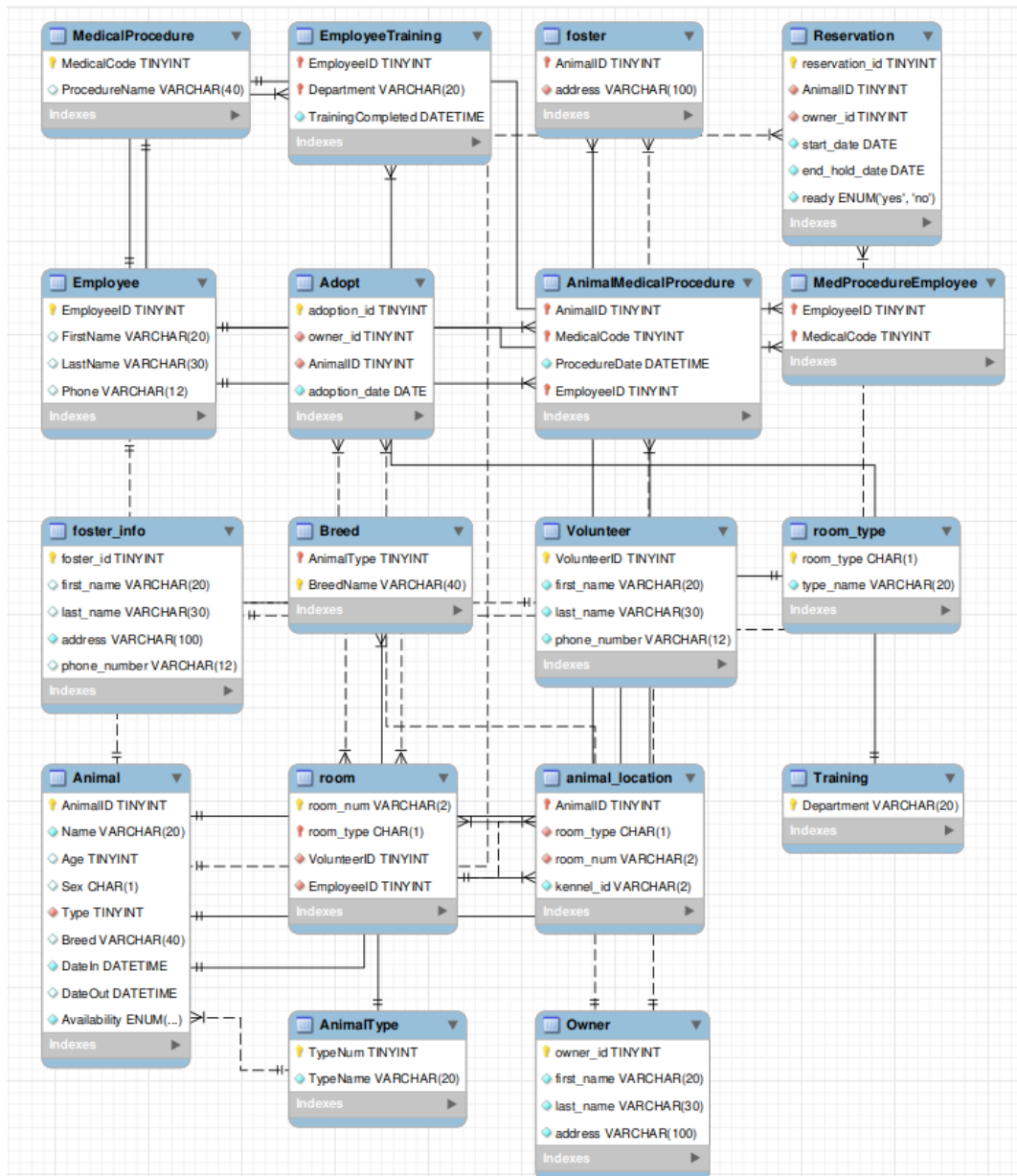
An animal shelter has two animal types it facilitates care and adoption for, cats or dogs. The animals receive various types of medical care in different rooms from employees who are trained to work in that department. Once the animals have received all necessary medical care many animals can be adopted or reserved by one owner. While waiting for adoption, they are

either in a room in a shelter being cared for by employees or volunteers, or they are in a foster home. A foster home can have many animals.

Attributes of:

- Animal: AnimalID, Name, Age, Sex, Type, Breed, DateIN, DateOut, and Availability
- MedicalProcedure: MedicalCode and ProcedureName
- EmployeeTraining: EmployeeID, Department, and TrainingCompleted
- Foster: AnimalID, Address
- Reservation: ReservationID, AnimalID, OwnerID, StartDate, EndHoldDate, and Ready
- Employee: EmployeeID, FirstName, LastName, and Phone
- Adopt: AdoptionID, OwnerID, AnimalID, and AdoptionDate
- AnimalMedicalProcedure: AnimalID, MedicalCode, ProcedureDate, and EmployeeID
- MedProcedureEmployee: EmployeeID and MedicalCode
- FosterInfo: FosterID, FirstName, LastName, Address, and PhoneNumber
- Breed: AnimalType and BreedName
- Volunteer: VolunteerID, FirstName, LastName, and PhoneNumber
- RoomType: TypeNum and TypeName
- Animal: AnimalID, Name, Age, Sex, Type, Breed, DateIn, DateOut, and Availability
- Room: RoomNum, RoomType, VolunteerID, EmployeeID, and KennelID
- AnimalType: TypeNum and TypeName
- Owner: OwnerID, FirstName, LastName, and Address
- AnimalLocation: AnimalID, RoomNum, RoomType, VolunteerID, EmployeeID, and KennelID

# ER Diagram



## ERD Description

1. Employee to EmployeeTraining(1:M)  
An employee can complete multiple training sessions, but each training session is associated with only one employee
2. Employee to MedProcedureEmployee(1:M)  
Each employee can perform multiple medical procedures, but each medical procedure record is tied to one employee
3. Employee to AnimalMedicalProcedure(1:M)  
An employee can be responsible for performing multiple medical procedures on animals, but each animal medical procedure is associated with only one employee
4. Animal to AnimalMedicalProcedure(1:M)  
Each animal can undergo multiple medical procedures, but each procedure is related to only one animal
5. Animal ↔ Adopt (1:M)  
Each animal can be adopted multiple times (though typically it will be adopted once in the real world), but each adoption record refers to one animal.
6. Animal ↔ Foster (1:M)  
An animal can be fostered by different people, but each fostering event refers to one animal.
7. Animal ↔ AnimalLocation (1:M)  
An animal can be in different locations (rooms, kennels) over time, but each record in the animal\_location table is for a single animal at a single location.
8. Breed ↔ Animal (1:M)  
A breed can have multiple animals, but each animal belongs to one breed.
9. Owner ↔ Adopt (1:M)  
An owner can adopt multiple animals, but each adoption record is linked to one owner.
10. Owner ↔ Animal (1:M)  
An owner can have multiple animals, but each animal is owned by only one owner.
11. Room ↔ AnimalLocation (1:M)  
A room can house multiple animals, but each animal\_location record corresponds to one animal in one room.
12. RoomType ↔ Room (1:M)  
A room type can be assigned to multiple rooms, but each room belongs to a single room type.
13. Training ↔ EmployeeTraining (1:M)  
Each training can be assigned to multiple employees, but each EmployeeTraining record refers to a single training.
14. Volunteer ↔ AnimalLocation (1:M)  
A volunteer can be assigned to many animal locations, but each animal location record corresponds to a single volunteer.
15. Volunteer ↔ Animal (1:M)



A volunteer can work with multiple animals, but each animal\_location record is linked to one animal and one volunteer.

16. Foster ↔ FosterInfo (1:1)

Each foster record has exactly one foster info entry, indicating a direct link between the foster and the fosterer's information.

## Approach Used to Develop Current Solution

To develop the current solution, we started by understanding the real-life needs of the Fur-ever Homes Adoption Center. We identified key processes such as animal intake, adoptions, medical care, room assignments, fostering and volunteer management. Based on these needs, we designed an organized database structure using entity-relationship diagrams (ERDs) to show how each table connects. We created tables with appropriate primary and foreign keys to reflect real-world relationships, such as which employee performed a medical procedure or which foster home an animal is assigned to. Throughout development, we focused on keeping the system easy to use, well-organized and flexible enough to support future updates like donation tracking or event scheduling.

## Current releases

Our first release is a working database system that includes all the main features needed. This version allows the shelter to store and manage information about animals, adopters, volunteers, medical care, and foster families. The system keeps track of whether each animal is available for adoption, still receiving care or placed in a foster home. It also includes a reservation feature, so interested adopters can reserve animals that are not yet ready to leave the shelter. Volunteers can be assigned to specific rooms or medical areas and foster families are linked to the animals they are temporarily caring for.

One new feature added in this release is the ability to track animal breeds. This helps adopters search for and choose the kind of cat or dog they are most interested in adopting. By including breed information in the database, the shelter can better match animals with adopters and make the adoption process more personal and effective. This release lays a strong foundation for a system that makes animal care and adoption smoother, faster and more organized.

## Future releases

In future releases, we plan to add new tables to improve the tracking and management of the shelter's operations. One idea is to introduce an Events table to record important shelter activities such as adoption events, medical checkup days, or volunteer training sessions. This will

help the shelter keep a history of events and manage scheduling better while allowing queries to track event attendance or animal participation.

Another useful addition could be a donation table to track donations made to the shelter, including donor details, donation amounts and dates. This information can support fundraising efforts and provide reports showing the shelter's financial support over time.

We also plan to create more complex queries, views, and stored procedures that make daily operations easier. For example, views could show animals needing follow up medical care or volunteers assigned to specific areas. Stored procedures might automate status updates for animals based on medical records or foster care duration. These improvements will increase efficiency and help shelter staff get important information quickly without changing the user interface.

## Conclusion

The Fur-ever Homes Adoption Center database project has created a complete, functional system to help the shelter better manage its animals, staff and day-to-day operations. With tables tracking animals, adoptions, fosters, volunteers, room assignments and medical care, the system mirrors the real-life needs of the shelter in a structured and organized way. Managers now have the tools to easily view and update information, make quick decisions about animal care or placement and ensure every animal receives the attention it needs. By including essential features such as reservation tracking, room availability and medical procedure logging, the database supports the shelter's goal of providing safe, caring and efficient services to animals and adopters.

Looking ahead, this database provides a strong foundation for future improvements. With plans to add features like event tracking, donation records and automated procedures, the system is designed to grow alongside the shelter's needs. The work done in this first release shows how thoughtful database design can reduce confusion, improve workflow and support the emotional and physical well being of every animal. As more features are added, this system has the potential to become an even more powerful tool for ensuring that no animal is overlooked and that every one finds its way to a fur-ever home.

## Appendix

### SQL Table Creation Scripts:

```
CREATE DATABASE Furever_Animal_Shelter; USE Furever_Animal_Shelter;
```

```
CREATE TABLE IF NOT EXISTS AnimalType( TypeNum TINYINT UNSIGNED NOT NULL, TypeName VARCHAR(20) NOT NULL, PRIMARY KEY (TypeNum) );
```

```
CREATE TABLE Breed ( AnimalType tinyint unsigned NOT NULL, BreedName varchar(40) NOT NULL, PRIMARY KEY (AnimalType,BreedName), CONSTRAINT Breed_ibfk_1 FOREIGN KEY (AnimalType) REFERENCES AnimalType (TypeNum) );
```

```
CREATE TABLE IF NOT EXISTS room_type( room_type CHAR(1) NOT NULL, type_name VARCHAR(20) NOT NULL, PRIMARY KEY (room_type));
```

```
CREATE TABLE Volunteer ( VolunteerID TINYINT UNSIGNED NOT NULL AUTO_INCREMENT, first_name VARCHAR(20) NOT NULL, last_name VARCHAR(30) NOT NULL, phone_number VARCHAR(12) NOT NULL, PRIMARY KEY (VolunteerID));
```

```
CREATE TABLE IF NOT EXISTS Employee( EmployeeID TINYINT UNSIGNED NOT NULL, FirstName VARCHAR(20), LastName VARCHAR(30), Phone VARCHAR(12), PRIMARY KEY (EmployeeID) );
```

```
CREATE TABLE IF NOT EXISTS Animal ( AnimalID TINYINT UNSIGNED NOT NULL AUTO_INCREMENT, Name VARCHAR(20) NOT NULL, Age TINYINT UNSIGNED, Sex CHAR(1), Type TINYINT UNSIGNED NOT NULL, Breed VARCHAR(40), DateIn DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP, DateOut DATETIME, Availability ENUM('AD', 'R', 'H', 'O') NOT NULL, PRIMARY KEY (AnimalID), FOREIGN KEY (Type) REFERENCES AnimalType(TypeNum));
```

```
CREATE TABLE room( room_num VARCHAR(2) NOT NULL, room_type CHAR(1) NOT NULL, VolunteerID TINYINT UNSIGNED NOT NULL, EmployeeID TINYINT UNSIGNED NOT NULL, PRIMARY KEY (room_type, room_num), FOREIGN KEY (room_type) REFERENCES room_type(room_type), FOREIGN KEY (VolunteerID) REFERENCES Volunteer(VolunteerID), FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID) );
```

```
CREATE TABLE animal_location ( AnimalID TINYINT UNSIGNED PRIMARY KEY, room_type CHAR(1) NOT NULL, room_num VARCHAR(2) NOT NULL, kennel_id VARCHAR(2) NOT NULL, FOREIGN KEY (AnimalID) REFERENCES Animal(AnimalID), FOREIGN KEY (room_type, room_num) REFERENCES room(room_type, room_num), UNIQUE (room_type, room_num, kennel_id));
```

```
CREATE TABLE IF NOT EXISTS Training( Department VARCHAR(20) NOT NULL, PRIMARY KEY (Department) );
```

```
CREATE TABLE IF NOT EXISTS EmployeeTraining( EmployeeID TINYINT UNSIGNED NOT NULL, Department VARCHAR(20) NOT NULL, TrainingCompleted DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP(), PRIMARY KEY (EmployeeID, Department),
```

FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID), FOREIGN KEY (Department) REFERENCES Training(Department) );

CREATE TABLE IF NOT EXISTS MedicalProcedure ( MedicalCode TINYINT UNSIGNED, ProcedureName VARCHAR(40), PRIMARY KEY (MedicalCode) );

CREATE TABLE IF NOT EXISTS MedProcedureEmployee( EmployeeID TINYINT UNSIGNED NOT NULL, ployeeID) REFERENCES Employee(EmployeeID), FOREIGN KEY (MedicalCode) REFERENCES MedicalProcedure(MedicalCode) );

CREATE TABLE IF NOT EXISTS AnimalMedicalProcedure( AnimalID TINYINT UNSIGNED NOT NULL, MedicalCode TINYINT UNSIGNED, ProcedureDate DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP ), EmployeeID TINYINT UNSIGNED NOT NULL, PRIMARY KEY (AnimalID, MedicalCode, EmployeeID), FOREIGN KEY(AnimalID) REFERENCES Animal(AnimalID), FOREIGN KEY (MedicalCode) REFERENCE S MedicalProcedure(MedicalCode), FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID) );

CREATE TABLE Owner ( owner\_id TINYINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY, first\_name VARCHAR(20) NOT NULL, last\_name VARCHAR(30) NOT NULL, address VARCHAR(100) NOT NULL);CREATE TABLE Reservation ( reservation\_id TINYINT UNSIGNED NOT NULL AUTO\_INCREMENT, AnimalID TINYINT UNSIGNED NOT NULL, owner\_id TINYINT UNSIGNED NOT NULL, sta rt\_date DATE NOT NULL, end\_hold\_date DATE NOT NULL, ready ENUM('yes', 'no') NOT NULL, alID), FOREIGN KEY (owner\_id) REFERENCES Owner(owner\_id) );

CREATE TABLE Adopt ( adoption\_id TINYINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY, owner\_id TINYINT UNSIGNED NOT NULL , AnimalID TINYINT UNSIGNED NOT NULL, adoption\_date DATE NOT NULL, FOREIGN KEY (owner\_id) REFERENCES Owner(owner\_id), FOREIGN KEY (AnimalID) REFERENCES Animal(AnimalID) );

CREATE TABLE foster\_info ( foster\_id TINYINT PRIMARY KEY, first\_name VARCHAR(20), last\_name VARCHAR(30), address VARCHAR(100) NOT NULL UNIQUE, phone\_number VARCHAR(12) );

CREATE TABLE foster( AnimalID TINYINT UNSIGNED PRIMARY KEY, -> address VARCHAR(100) NOT NULL UNIQUE, FOREIGN KEY (address) REFERENCES foster\_info(address), FOREIGN KEY (AnimalID) REFERENCES Animal(AnimalID));

## SQL Data Population Scripts:

```
INSERT INTO AnimalType(TypeNum, TypeName) VALUES (1,'Cat'), (2, 'Dog');
```

```
INSERT INTO Breed(AnimalType, BreedName) VALUES (1,'Unknown'), (1,'American Shorthair'), (1, 'Norwegian Forest'), (1, 'Siamese'), (1,'Ragdoll'), (2, 'Unknown'), (2, 'Labrador Retriever'), (2, 'Chihuahua'), (2, 'Poodle'), (2, 'Bulldog'), (2, 'Australian Shepherd');
```

```
INSERT INTO `Adopt` VALUES (1,1,1,'2025-08-11'),(2,2,2,'2025-08-01'),(3,3,3,'2025-08-06'),(4,4,4,'2025-08-03'),(5,5,5,'2025-08-04');
```

```
INSERT INTO `Animal` VALUES (1,'Milo',2,'M',1,'American Shorthair','2025-08-06 00:49:38',NULL,'AD'),(2,'Luna',3,'F',2,'Labrador Retriever','2025-08-06 00:49:38',NULL,'H'),(3,'Bella',1,'F',1,'Siamese','2025-08-06 00:49:38',NULL,'R'),(4,'Max',4,'M',2,'Poodle','2025-08-06 00:49:38',NULL,'AD'),(5,'Daisy',5,'F',2,'Bulldog','2025-08-06 00:49:38',NULL,'O');
```

```
INSERT INTO `AnimalMedicalProcedure` VALUES (1,10,'2025-08-01 00:00:00',1),(2,20,'2025-08-02 00:00:00',2),(3,30,'2025-08-03 00:00:00',3),(4,10,'2025-08-04 00:00:00',4),(5,20,'2025-08-05 00:00:00',5);
```

```
INSERT INTO `Employee` VALUES (1,'Sarah','Nguyen','5551112222'),(2,'David','Kim','5552223333'),(3,'Riley','Martinez','5553334444'),(4,'Chris','Thompson','5554445555'),(5,'Pat','Lopez','5555556666');
```

```
INSERT INTO `EmployeeTraining` VALUES (1,'Intake','2025-08-06 00:54:57'),(2,'Vaccinations','2025-08-06 00:54:57'),(3,'Surgical','2025-08-06 00:54:57'),(4,'Nursery','2025-08-06 00:54:57');
```

```
INSERT INTO `MedProcedureEmployee` VALUES (1,10),(4,10),(2,20),(5,20),(3,30);
```

```
INSERT INTO `MedicalProcedure` VALUES (10,'General Checkup'),(20,'Vaccinations'),(30,'Spay / Neuter');
```

```
INSERT INTO `Owner` VALUES (1,'Emily','Stone','101 Maple St'),(2,'John','Doe','202 Oak Ave'),(3,'Luna','West','303 Pine Blvd'),(4,'Nina','Bell','404 Cedar Ct'),(5,'Ethan','Gray','505 Birch Rd');
```

```
INSERT INTO `Reservation` VALUES (1,1,1,'2025-08-01','2025-08-10','yes'),(2,2,2,'2025-07-20','2025-07-30','no'),(3,3,3,'2025-08-05','2025-08-12','yes'),(4,4,4,'2025-08-02','2025-08-09','no'),(5,5,5,'2025-08-03','2025-08-11','yes');
```

INSERT INTO `Training` VALUES ('Intake'),('Nursery'),('Surgical'),('Vaccinations');

INSERT INTO `Volunteer` VALUES

(1,'Alex','Rivera','5551234567'),(2,'Jamie','Lee','5552345678'),(3,'Morgan','Smith','5553456789'),(4,'Taylor','Brown','5554567890'),(5,'Jordan','Davis','5555678901');

INSERT INTO `foster` VALUES (1,'1 Foster Ln'),(2,'2 Foster Ln'),(3,'3 Foster Ln'),(4,'4 Foster Ln'),(5,'5 Foster Ln')

INSERT INTO `foster\_info` VALUES (1,'Anna','Ford','1 Foster Ln','5551110001'),(2,'Tom','Rogers','2 Foster Ln','5552220002'),(3,'Cathy','Long','3 Foster Ln','5553330003'),(4,'Ben','Hill','4 Foster Ln','5554440004'),(5,'Nora','Stone','5 Foster Ln','5555550005');

INSERT INTO `room` VALUES

('04','M',4,4),('03','N',3,3),('05','N',5,5),('01','S',1,1),('02','S',2,2);

INSERT INTO animal\_location (AnimalID, room\_type, room\_num, kennel\_id) VALUES (4, 'S', '01', 'A'), (2, 'S', '01', 'B'), (3, 'S', '02', 'A'), (1, 'N', '03', 'A'), (5, 'S', '01', 'C');

INSERT INTO `room\_type` VALUES ('M','Medical Room'),('N','Nursery'),('S','Standard Room');

INSERT INTO `animal\_location` VALUES

(1,'N','03','A'),(4,'S','01','A'),(2,'S','01','B'),(5,'S','01','C'),(3,'S','02','A');

## Time Log

Date	Team Member	Hours Spent	Description of work	Additional Comments
15 July 2025	Alissia	3	Brainstormed what project we want to work on, made conceptual design and made the proposal.	
15 July 2025	Sofia	3	Brainstormed what project we want to work on, made conceptual design and made the proposal.	

15 July 2025	Bishakha	3	Brainstormed what project we want to work on, made conceptual design and made the proposal.	
25 July 2025	Alissia	3	Worked on erd and what tables we are going to build.	Talked to professor craig and made some changes on our design since the first meeting.
25 July 2025	Sofia	3	Worked on erd and what tables we are going to build.	Talked to professor craig and made some changes on our design since the first meeting.
25 July 2025	Bishakha	3	Worked on erd and what tables we are going to build.	Talked to professor craig and made some changes on our design since the first meeting.
27 July 2025	Alissia	3	Worked on the tables we divided and added some values.	
27 July 2025	Sofia	3	Worked on the tables we divided and added some values.	

27 July 2025	Bishakha	2.5	Worked on the tables we divided and added some values.	
30 July 2025	Alissia	2	Tried to see if our tables worked together and fixed some relationship errors after consulting professor Craig.	
30 July 2025	Sofia	2	Tried to see if our tables worked together and fixed some relationship errors after consulting professor Craig.	
30 July 2025	Bishakha	2	Tried to see if our tables worked together and fixed some relationship errors after consulting professor Craig.	
5 August 2025	Alissia	6	Fixed some errors on table values, worked on the final report and made slideshows.	
5 August 2025	Sofia	6	Created the database and ran the create and insert and fixed some relationship errors, worked on the final report.	



5 August 2025	Bishakha	6	Worked on the final report.	
6 August 2025	Alissia	2	Worked on queries and completed the slideshow and the final report.	
6 August 2025	Sofia	2	Worked on queries and completed the slideshow and the final report.	
6 August 2025	Bishakha	2	Worked on queries and completed the slideshow and the final report.	

1. Looking for a dog a 3 year old Lab retriever dog's availability

```
mysql> SELECT Animal.Name, Animal.Availability FROM Animal WHERE Animal.Age = 3
& Animal.Breed = 'Labrador Retriever'^C
mysql> SELECT Animal.Name, Animal.Availability FROM Animal WHERE Animal.Age = 3
AND Animal.Breed = 'Labrador Retriever';
+-----+-----+
| Name | Availability |
+-----+-----+
| Luna | H           |
+-----+-----+
1 row in set (0.001 sec)
```

2. Which animals have been in the shelter the longest?

```
mysql> SELECT AnimalID, Name, DateIn
-> FROM Animal
-> WHERE DateOut IS NULL
-> ORDER BY DateIn ASC;
```

```
+-----+-----+-----+
| AnimalID | Name | DateIn |
+-----+-----+-----+
| 1 | Milo | 2025-08-06 00:49:38 |
| 2 | Luna | 2025-08-06 00:49:38 |
| 3 | Bella | 2025-08-06 00:49:38 |
| 4 | Max | 2025-08-06 00:49:38 |
| 5 | Daisy | 2025-08-06 00:49:38 |
+-----+-----+-----+
5 rows in set (0.001 sec)
```

3. The adopter asks for a Siamese cat that is under the age of 2 and also list out if it has any medical procedures and the date of the pr

```
mysql> SELECT Animal.AnimalID AS AnimalID, Animal.Name AS AnimalName, Animal.Age, MedicalProcedure.ProcedureName AS MedicalProcedure, AnimalMedicalProcedure.ProcedureDate FROM Animal LEFT JOIN AnimalMedicalProcedure ON Animal.AnimalID = AnimalMedicalProcedure.AnimalID LEFT JOIN MedicalProcedure ON AnimalMedicalProcedure.MedicalCode = MedicalProcedure.MedicalCode WHERE Animal.Breed = 'Siamese' AND Animal.Age < 2 AND Animal.Type = 1;
```

```
+-----+-----+-----+-----+-----+
| AnimalID | AnimalName | Age | MedicalProcedure | ProcedureDate |
+-----+-----+-----+-----+-----+
| 3 | Bella | 1 | Spay / Neuter | 2025-08-03 00:00:00 |
+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

4. who is working in what room and their trainings

```
mysql> SELECT Employee.FirstName, Employee.LastName, room.room_num, room.room_type, EmployeeTraining.Department, EmployeeTraining.TrainingCompleted FROM room JOIN Employee ON room.EmployeeID = Employee.EmployeeID JOIN EmployeeTraining ON Employee.EmployeeID = EmployeeTraining.EmployeeID;
```

```
+-----+-----+-----+-----+-----+-----+
| FirstName | LastName | room_num | room_type | Department | TrainingCompleted |
+-----+-----+-----+-----+-----+-----+
| Sarah | Nguyen | 01 | S | Intake | 2025-08-06 00:54:57 |
| David | Kim | 02 | S | Vaccinations | 2025-08-06 00:54:57 |
| Riley | Martinez | 03 | N | Surgical | 2025-08-06 00:54:57 |
| Chris | Thompson | 04 | M | Nursery | 2025-08-06 00:54:57 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```