Jorge Santana Prof. Peralta Ramos

CIS 344 9/6/2025

**PROJECT SELECTION:**

For my Mini World I choose Pet Training School because I like domestic animals and I think it would be fun and engaging to design a database based on a pet training school.

**SYSTEM DESIGN REQUIREMENTS:**

To decide what information my Pet Training School system should include, I looked at a few helpful resources. The first one was the [*Dog Academy*](https://dogacademy.org/) website. It explained the types of training programs offered for different kinds of pets, which helped me see that I needed to include tables for **owners**, **pets**, and **courses**. It also gave me the idea to include details such as the pet’s species and breed, since training can differ between animals.

Next, I reviewed the [*Animal Behavior College*](https://www.animalbehaviorcollege.com/) website. This site showed how professional trainers structure courses and training sessions. From this, I realized I should include a **trainer** table with details about their specialties. I also saw the importance of tracking which pets are enrolled in which courses, so I created an **enrollment** table that links pets, courses, and trainers together.

Finally, I watched YouTube videos about how pet training schools operate. These explained the step-by-step process of enrolling a pet, attending sessions, and completing courses. This confirmed that my system should be able to store enrollment status such as *Enrolled* or *Completed*.

From these sources, I decided that my system must store information about pet owners, their pets, available courses, trainers, and the enrollment process. I also decided that foreign keys would be necessary to connect pets with their owners, and to link pets to courses and trainers through enrollment records.

My Pet Training School entities are:

* **Owner** (OwnerID, FirstName, LastName, Phone, Email)
* **Pet** (PetID, OwnerID, PetName, Species, Breed) OwnerID is a foreign key.
* **Trainer** (TrainerID, FirstName, LastName, Specialty)
* **Course** (CourseID, Title, Description, Price)
* **Enrollment** (EnrollmentID, PetID, CourseID, TrainerID, Status) PetID, CourseID, and TrainerID are foreign keys.

**ENTITY-RELATIONSHIP (ER) DIAGRAMS:**

In the Chen-style ER diagram, I represented each of the five entities (Owner, Pet, Trainer, Course, Enrollment) and illustrated the relationships: an Owner can have many Pets, Pets can enroll in multiple Courses, and each Enrollment is linked to a specific Trainer. In MySQL Workbench, I recreated the design using UML notation, ensuring that primary and foreign key relationships were properly represented.

**DATABASE CREATION:**

After creating the UML diagram in MySQL Workbench, I used the Forward Engineer tool to generate the SQL script (PetTrainingSchool.sql) and created the database. This process automatically created the tables and their foreign key relationships on my local MySQL server.

In conclusion, this project helped me understand how to analyze real-world requirements, translate them into entities and relationships, and implement them in MySQL Workbench. One challenge was determining the right attributes for each entity, but using multiple sources helped me refining my design. Another challenge was determining whether the connection was one-to-many, one-to-one, or many-to-many. Overall, the project gave me firsthand experience in database modeling and SQL development.