Gin Sang

gbs0116@mavs.uta.edu

1001810116 (02/16/24)

Term Project: Phase I, II , & III

Pet Adoption Database

Step 1: Consider What To Do

I am choosing to do a PET ADOPTION AGENCY database for my miniworld. In this database, we will be managing pets, owners, breeds, adoptions, locations, and veterinarians as entity types and their relationships

Step 2: Problem Definition, User Requirements

There is a need for this pet adoption agency, because approximately 70% of the population owns a pet, and unfortunately this results in many pets that no longer wanted and left to fend for themselves. This Pet Adoption Agency works to solve this issue by enabling a database system that manages the pet adoption process. Users can create queries that are appropriate to their needs which will make the adoption process more efficient. The database will store information about pets that are available for adoption and create relationships with other entities such as the location of the pets that are available, the closest veterinarian, the adoption records, the breed of each pet, etc. Each PET has a unique identifier. Each PET is stationed at one LOCATION. Each PET belongs to one OWNER. Each PET is treated by one VETERINARIAN. Each VETERINARIAN can treat multiple PETS. Each PET is a member of one or more BREED. Each PET has ADOPTION information. Each VETERINARIAN is stationed at one LOCATION

Step 3: 10 Realistic Queries (Some Queries have been changed in order to make it more realistic, this is the UPDATED QUERIES)

1. Retrieve names, ages, breeds, and locations of all available pets.

   Entities involved: PET, BREED, LOCATION

2. Retrieve names, countries of origins, breeds, and adoption fees of available pets.

   Entities involved: PET, BREED, ADOPTION

3. Retrieve owner's name, contact information, adoption date, and adoption location for a specific pet.

   Entities involved: PET, OWNER, LOCATION, ADOPTION

4. Find the average age of pets, grouped by breed, filtering out breeds with an average age greater than 2, ordered by the average age in descending order.

   Entities involved: PET, BREED

5. Find an available pet that is 2 years of age, that is a labrador retriever, with an adoption fee that is less than 200 dollars.

   Entities involved: PET, ADOPTION

6. Find the average age of pets, grouped by breed, only considering breeds with an average lifespan greater than 10 years, ordered by the average age in descending order.

   Entities involved: PET, BREED

7. Retrieve the names and ages of all pets that are currently unavailable.

   Entities involved: PET

8. Retrieve the names, breeds, and average lifespans of all breeds along with the count of pets for each breed.

   Entities involved: PET, BREED

9. Find the total number of pets available in each location.

   Entities involved: PET, LOCATION

10. Retrieve the names and contact information of owners who adopted pets in the month of May 2023.

    Entities involved: OWNER, ADOPTION

Step 4: Entity-Relationship with assumptions (Chapter 3)

PET entity type

Attributes:

Name(String)

Age(Integer)

Breed_ID(Integer)

Availability_Status (Boolean)

Date_of_Arrival(Date)

Location_ID(Integer)

OWNER entity type

Attributes:

Owner_ID(Integer)

Name(String)

Address(String)

Contact_Information(String)

BREED entity type

Attributes:

Breed_ID(Integer)

Name(String)

Country_of_Origin(String)

Average_Lifespan(Integer)

ADOPTION

Attributes:

Adoption_ID(Integer)

Pet_ID(Integer)

Owner_ID(Integer)

Adoption_Date(String)

Adoption_Fee(Float)

Location_ID(Integer)

VETERINARIAN

Attributes :

Veterinarian_ID(Integer)

Name(String)

Specialization(String)

Contact_Information(String)

LOCATION

Attributes:

Location_ID(Integer)

City(String) State(String)

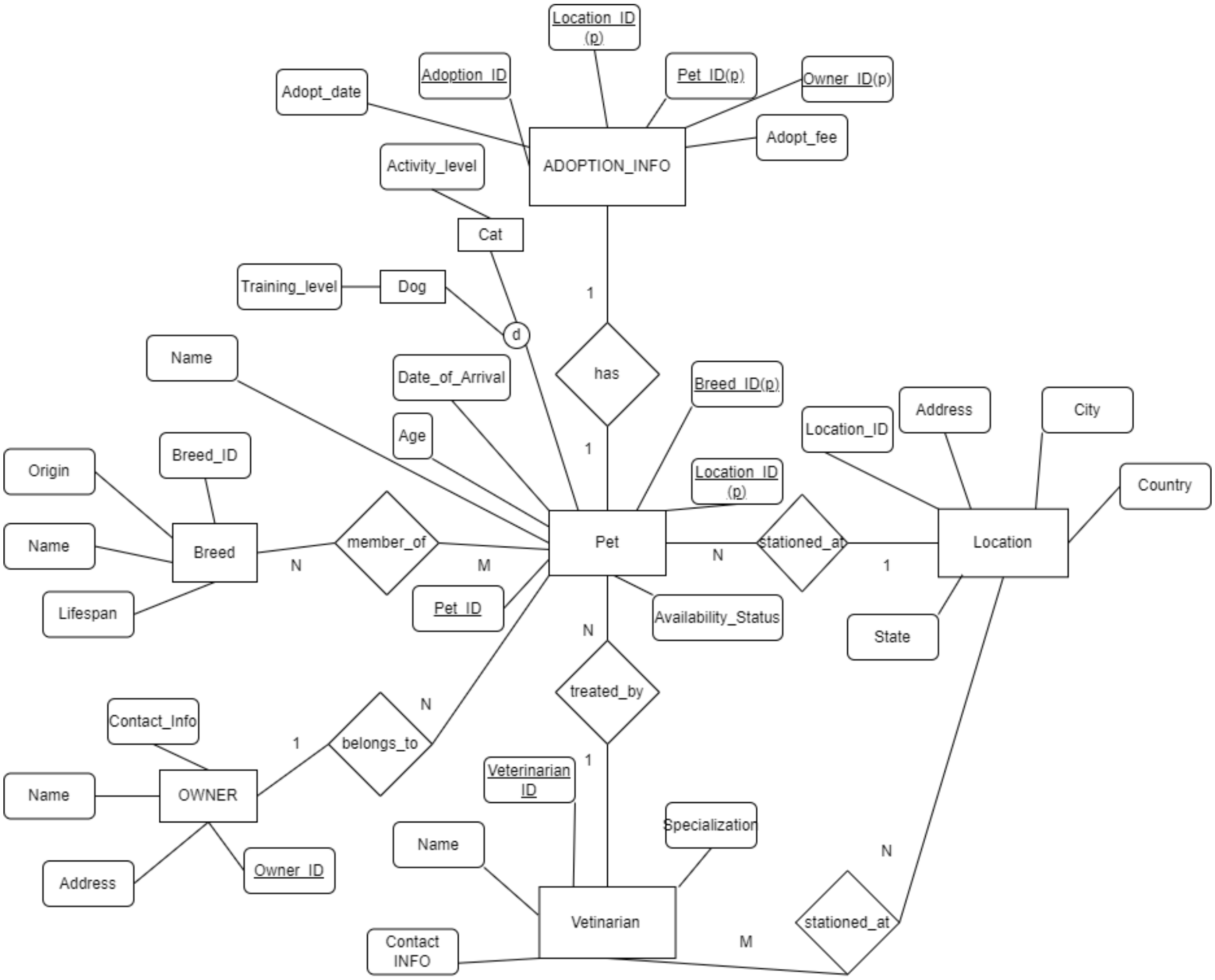
Country(String)

Address(String)

Cardinality Stationed_at relationship type: PET is stationed_at one LOCATION Cardinality is 1:1 (Each pet is stationed at one location) Works_For relationship type: EMPLOYEE Works_For DEPARTMENT Cardinality is N:1 (Multiple employees work for one department) Belongs_to relationship type: PET Belongs_to OWNER Cardinality is N:1 (Many pets can belong to one owner) Treated_by relationship type: PET Treated_by VETERINARIAN Cardinality is N:1 (Many pets can be treated by one veterinarian) Stationed_at relationship type: VETERINARIAN Stationed_at LOCATION Cardinality is 1:N (One veterinarian is stationed at one location, but one location can have multiple veterinarians.) Belongs_to relationship type: PET Belongs_to BREED Cardinality is N:M (Many pets can belong to one or more breeds, and each breed can have one or more pets) Has_information relationship type: PET Has_information about their ADOPTION Cardinality is 1:1 (Each pet has adoption information, and each adoption information corresponds to one pet)

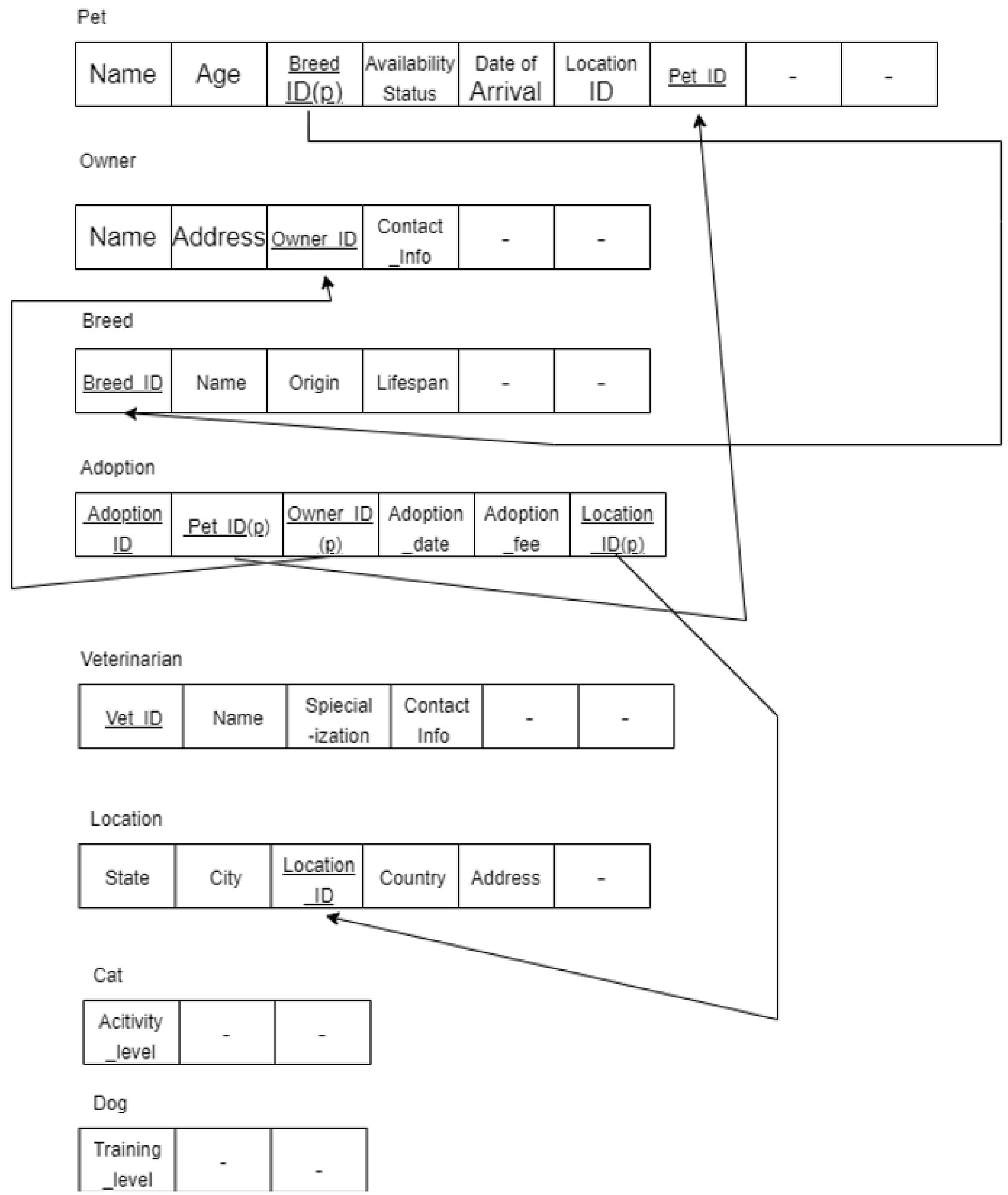
Assumptions: In case of a mutt (A dog that is of multiple breeds), the name of the breed will just be all of the breed that is made out of. For example if it is a mutt crossed between a maltese and a poodle. For Breed ID it should have both ID for the maltese and poodle. Conclusion: From this activity, I learned to create entity types, and show the relationships between each entity. I also learned to create attributes for these entities. I learned the importance of creating the entity and their relationships first in order to come up with 10 realistic queries.

Step 5: EER modeling



*** Changes: Whenever there is the "(p)" I meant to say foreign key "(fk)" . I totally mixed up foreign keys with partial keys.**

Step 6: Creating Relations



Pet

Name	Age	Breed ID(p)	Availability Status	Date of Arrival	Location ID	Pet_ID	-	-

Owner

Name	Address	Owner_ID	Contact _Info	-	-

Breed

Breed_ID	Name	Origin	Lifespan	-	-

Adoption

Adoption ID	Pet_ID(p)	Owner_ID (p)	Adoption _date	Adoption _fee	Location _ID(p)

Veterinarian

Vet_ID	Name	Spiecial -ization	Contact Info	-	-

Location

State	City	Location _ID	Country	Address	-

Cat

Acitivity _level	-	-

Dog

Training _level	-	-

*** The boxes with blank are just empty they aren’t incomplete. However for Cat and Dog.**

These are it’s new attributes:

```
INSERT INTO Dog (Dog_ID, Color, Size, Temperament, Health_Status) VALUES
(101, 'Brown', 'Medium', 'Friendly', 'Healthy'),
(102, 'Black', 'Large', 'Energetic', 'Healthy'),
(104, 'White', 'Small', 'Playful', 'Healthy'),
(105, 'Golden', 'Medium', 'Loyal', 'Healthy'),
(110, 'Gray', 'Large', 'Intelligent', 'Healthy');

-- sample data into Cat table
INSERT INTO Cat (Cat_ID, Color, Size, Temperament, Health_Status) VALUES
(103, 'White', 'Small', 'Independent', 'Healthy'),
(106, 'Gray', 'Medium', 'Calm', 'Healthy'),
(107, 'Black', 'Small', 'Affectionate', 'Healthy'),
(108, 'Tabby', 'Medium', 'Curious', 'Healthy'),
(109, 'Calico', 'Medium', 'Sweet', 'Healthy');
```

***It proves to be too time consuming for me to edit due to not having the EER model saved, so I'm just manually explaining the changes.**

Step 7: Relational Algebra of OLD queries. Some of these queries have now been replaced. (See step 3)

1. Get the names, ages, and breeds of all pets that are currently available, along with where each pet is located. Entities involved: PET, BREED,LOCATION

π(Name, Age, Breed, City, State, Country)(σ Availability_Status = True (PET $\bowtie$ LOCATION))

2. Make a list of available pets with their names, countries of origins, their breed, and adoption fee. Entities involved: PET, BREED,ADOPTION

π(Name, Country_of_Origin, Breed, Adoption_fee)(σ Availability_Status = True (PET $\bowtie$ BREED $\bowtie$ ADOPTION))

3. For a specific pet, retrieve the owner's name, their contact information, the adoption date, and the location at which the pet was adopted. Entities involved: PET, OWNER, LOCATION

π(Owner_Name, Contact_Information, Adoption_Date, City, State, Country)(σPET.Name = ‘specific_pet_name(PET ⋈ OWNER ⋈LOCATION⋈ADOPTION))

4. List the names, ages and specializations of all pets treated by each veterinarian, along with the location at which the pet was adopted. Entities involved: PET, VETERINARIAN, LOCATION

π(Name, Age, Specialization, City, State, Country)(PET ⋈ VETERINARIAN ⋈ ADOPTION ⋈ LOCATION)

5. Find an available pet that is 9 years of age, that is a shih tzu, with an adoption fee that is less than 50 dollars. Entities involved: PET, OWNER, ADOPTION

π(Name, Age, Breed, City, State, Country)(σ Age = 9 ^ Breed = ‘Shih Tzu’ ^ Adoption _ Fee < 50 (PET ⋈ LOCATION ⋈ BREED ⋈ ADOPTION))

6. Find an available pet that is of the breed bulldog, and a veterinarian located in Tennessee that specializes in bulldogs. Entities involved: PET, VETERINARIAN, LOCATION

π(Name, Age, Breed, City, State, Country)(σ Breed = ‘Bulldog’ ^ State = ‘TN’ ^ Specialization = ‘Bulldog’ (PET ⋈ VETERINARIAN ⋈ LOCATION ⋈ BREED ⋈ ADOPTION))

7. Name a list of owners and their contact information, that recently adopted a pet in the last 30 days, and the names of the pet they adopted. Entities involved: PET, OWNER, ADOPTION

π(Owner_Name, Contact_Informaiton, Pet_Name(σ Adoption_Date >= current_date - 30 (PET ⋈ OWNER ⋈ ADOPTION))

8. Estimate the distance of an owner and the nearest veterinarian. Entities involved: OWNER, VETERINARIAN, LOCATION

Estimate distance using appropriate distance calculation function (OWNER ⋈ LOCATION ⋈ VETERINARIAN⋈ LOCATION)

9. How long does the average pet take to get adopted, and what is the average age of dogs that are adopted, and the most common breed? Entities involved: PET, ADOPTION, BREED

Use aggregate functions to find average adoption time, average age of adopted dogs, and most common breed(ADOPTION ⋈PET)

10. List information of all pets that are available for adoption, with an adoption fee between $50-100, and list them in the order of the closest veterinarian. Entities involved: PET, ADOPTION, VETERINARIAN

Order pets by distance and filter by adoption fee between 50 and 100 , and list them in the order of the closest veterinarian(PET ⋈ ADOPTION ⋈ VETERINARIAN ⋈ LOCATION).

What I have learned Conclusion:

Problem Definition and User Requirements: Recognizing the need for a pet adoption agency database due to the high number of pets in need of homes, I identified key user requirements to streamline the adoption process and improve efficiency.

1. Entity-Relationship Modeling: I developed an understanding of how to model entities, attributes, and relationships, which formed the foundation of the database design. By carefully defining these elements, I ensured that the database accurately represented the real-world scenario of a pet adoption agency.

2. Realistic Queries: Through the creation of realistic queries, I learned how to translate user requirements into actionable database queries. These queries ranged from simple retrievals to more complex analyses, providing a comprehensive view of the database's capabilities.
3. EER Modeling and Schema Design: Utilizing EER modeling techniques, I visualized the structure of the database and designed an appropriate schema to capture all relevant information. This step was crucial in ensuring data integrity and efficiency in storage and retrieval.
4. Creating Relations: I successfully established relationships between entities, considering cardinality and constraints to accurately represent the connections between different parts of the database.
5. Relational Algebra: By applying relational algebra, I translated database queries into mathematical operations, further solidifying my understanding of query execution and optimization.
6. Learning from Challenges: Throughout the project, I encountered challenges such as handling mutt breeds and maintaining data consistency. These challenges provided valuable learning opportunities, teaching me to approach problems systematically and find effective solutions.
7. One of the significant challenges I encountered was ensuring seamless communication between the frontend and backend systems. Integrating the HTML and CSS frontend with the MySQL database involved overcoming various technical complexities. Configuring the system settings to enable smooth interaction between different components demanded careful consideration and thorough testing. Despite these challenges, I tackled each

obstacle with determination and perseverance, utilizing research, experimentation, and collaboration with the GTA to overcome hurdles and make progress in the project. Each challenge presented an opportunity for growth, enhancing my skills in web development, database management, and system administration.