Capital University of Science and Technology

Department of Software Engineering

COURSE TITLE: DATABASE

ASSIGNMENT 03

Semester: FALL 2024 Max Marks: 10

Name: Wajeeha Naeem (BSE233118)
Yahya Sami (BSE233114)
Maheen Gul (BSE233113)

Instructor: Hina Rashid

Date: 27/12/2024

GitHub Links

Wajeeha Naeem: https://github.com/wajeeha51

Yahya Sami: https://github.com/YahyaSami1147

Maheen Gul: https://github.com/Maheen1147

Animal Shelter Management System: Conceptual and Logical Design

1. Description of Entities, Attributes, and Relationships

Entities and Attributes:

1. **Animals:**
    - **Primary Key:** animalID
    - **Attributes:** name, species, age, medicalHistory, dietaryRequirements
2. **Dogs:**
    - **Primary Key:** animalID (foreign key to Animals)
    - **Attributes:** breed, size, vaccinationStatus, groomingNeeds, temperament
3. **Cats:**
    - **Primary Key:** animalID (foreign key to Animals)
    - **Attributes:** breed, color, vaccinationStatus, personalityTraits, groomingNeeds
4. **ShelterEmployees:**
    - **Primary Key:** employeeID
    - **Attributes:** name, role, hireDate, contactInformation, schedule
5. **AdoptionRequests:**
    - **Primary Key:** requestID
    - **Attributes:** adopterName, animalType, adoptionDate, status, approvalDate
6. **Adopters:**
    - **Primary Key:** adopterID
    - **Attributes:** name, contactInformation, address, employmentStatus, previousPets

Relationships:

1. **Animals - Dogs:** One-to-One
    - One animal can be a dog.
    - Dogs' animalID references Animals' animalID.
2. **Animals - Cats:** One-to-One
    - One animal can be a cat.
    - Cats' animalID references Animals' animalID.

3. **ShelterEmployees - AdoptionRequests:** One-to-Many
    - One shelter employee handles many adoption requests.
    - AdoptionRequests' employeeID references ShelterEmployees' employeeID.
4. **AdoptionRequests - Animals:** One-to-Many
    - One adoption request refers to one animal.
    - AdoptionRequests' animalID references Animals' animalID.
5. **Adopters - AdoptionRequests:** One-to-Many
    - One adopter can make multiple requests.
    - AdoptionRequests' adopterID references Adopters' adopterID.

2. Challenges and Considerations

1. **Hierarchical Relationships:** Defining Dogs and Cats as subcategories of Animals while avoiding redundancy required careful use of foreign keys.
2. **Entity Relationships:** Ensuring that each AdoptionRequest links correctly to both an animal and an adopter, while maintaining consistency.
3. **Data Integrity:** Deciding on appropriate constraints to prevent orphaned records (e.g., ensuring that an AdoptionRequest always links to a valid animal).

3. Tables, Fields, and Data Types

Table Name	Field Name	Data Type
Animals	animalID	INT (Primary Key)
	name	VARCHAR(50)
	species	VARCHAR(20)
	age	INT
	medicalHistory	TEXT
	dietaryRequirements	TEXT
Dogs	animalID	INT (Foreign Key)

Table Name	Field Name	Data Type
	breed	VARCHAR(30)
	size	VARCHAR(15)
	vaccinationStatus	VARCHAR(15)
	groomingNeeds	VARCHAR(50)
	temperament	VARCHAR(50)
Cats	animalID	INT (Foreign Key)
	breed	VARCHAR(30)
	color	VARCHAR(20)
	vaccinationStatus	VARCHAR(15)
	personalityTraits	VARCHAR(50)
	groomingNeeds	VARCHAR(50)
ShelterEmployees	employeeID	INT (Primary Key)
	name	VARCHAR(50)
	role	VARCHAR(30)
	hireDate	DATE
	contactInformation	VARCHAR(100)
	schedule	VARCHAR(50)
AdoptionRequests	requestID	INT (Primary Key)
	adopterName	VARCHAR(50)
	animalType	VARCHAR(20)
	adoptionDate	DATE
	status	VARCHAR(20)

Table Name	Field Name	Data Type
	approvalDate	DATE
Adopters	adopterID	INT (Primary Key)
	name	VARCHAR(50)
	contactInformation	VARCHAR(100)
	address	TEXT
	employmentStatus	VARCHAR(20)
	previousPets	VARCHAR(100)

4. Establishing Relationships

- **Animals' animalID** serves as a foreign key in Dogs and Cats.
- **AdoptionRequests' employeeID** is a foreign key referencing ShelterEmployees' employeeID.
- **AdoptionRequests' animalID** is a foreign key referencing Animals' animalID.
- **AdoptionRequests' adopterID** is a foreign key referencing Adopters' adopterID.

5.Transformation of ERD into Relational Data Model (RDM)

To translate the conceptual ERD into a relational data model, we mapped entities, attributes, and relationships into well-structured relational tables. Below are the relational tables derived from the ERD, including their primary and foreign keys. The relational model ensures normalization and maintains data integrity.

Relational Tables

1. **Animals Table**
    - **Primary Key:** animalID
    - **Attributes:** name, species, age, medicalHistory, dietaryRequirements

Field Name	Data Type	Constraint
animalID	INT	PRIMARY KEY
name	VARCHAR(50)	NOT NULL
species	VARCHAR(20)	NOT NULL
age	INT	
medicalHistory	TEXT	
dietaryRequirements	TEXT	

2. **Dogs Table**
    - **Primary Key:** animalID (foreign key to Animals)
    - **Attributes:** breed, size, vaccinationStatus, groomingNeeds, temperament

Field Name	Data Type	Constraint
animalID	INT	PRIMARY KEY, FOREIGN KEY (Animals)
breed	VARCHAR(30)	NOT NULL
size	VARCHAR(15)	
vaccinationStatus	VARCHAR(15)	
groomingNeeds	VARCHAR(50)	
temperament	VARCHAR(50)	

3. **Cats Table**
    - **Primary Key:** animalID (foreign key to Animals)
    - **Attributes:** breed, color, vaccinationStatus, personalityTraits, groomingNeeds

Field Name	Data Type	Constraint
animalID	INT	PRIMARY KEY, FOREIGN KEY (Animals)
Breed	VARCHAR(30)	NOT NULL
Color	VARCHAR(20)	
vaccinationStatus	VARCHAR(15)	

Field Name	Data Type	Constraint
personalityTraits	VARCHAR(50)	
groomingNeeds	VARCHAR(50)	

4. **ShelterEmployees Table**
    - **Primary Key:** employeeID
    - **Attributes:** name, role, hireDate, contactInformation, schedule

Field Name	Data Type	Constraint
employeeID	INT	PRIMARY KEY
Name	VARCHAR(50)	NOT NULL
role	VARCHAR(30)	
hireDate	DATE	
contactInformation	VARCHAR(100)	
schedule	VARCHAR(50)	

5. **Adopters Table**
    - **Primary Key:** adopterID
    - **Attributes:** name, contactInformation, address, employmentStatus, previousPets

Field Name	Data Type	Constraint
adopterID	INT	PRIMARY KEY
name	VARCHAR(50)	NOT NULL
contactInformation	VARCHAR(100)	
address	TEXT	
employmentStatus	VARCHAR(20)	
previousPets	VARCHAR(100)	

6. **AdoptionRequests Table**
    - **Primary Key:** requestID

- **Foreign Keys:** adopterID (Adopters), employeeID (ShelterEmployees), animalID (Animals)
- **Attributes:** adopterName, animalType, adoptionDate, status, approvalDate

Field Name	Data Type	Constraint
requestID	INT	PRIMARY KEY
adopterID	INT	FOREIGN KEY (Adopters)
employeeID	INT	FOREIGN KEY (ShelterEmployees)
animalID	INT	FOREIGN KEY (Animals)
adopterName	VARCHAR(50)	
animalType	VARCHAR(20)	
adoptionDate	DATE	
status	VARCHAR(20)	
approvalDate	DATE	

Relational Model

6.Report on Design Decisions

The design of the relational data model for the Animal Shelter Management System involved balancing normalization, scalability, and ease of use. Below are the major considerations and decisions taken during the process:

1. Normalization

- The schema was normalized to the third normal form (3NF) to eliminate redundancy and ensure data consistency.
- For example, Dogs and Cats inherit their animalID from the Animals table, separating general attributes like name and species from specific attributes like breed and groomingNeeds.

2. Referential Integrity

- Foreign keys were implemented to establish relationships between tables, such as linking AdoptionRequests to Animals and Adopters.
- This ensures that no orphaned records exist and allows for cascading updates or deletes.

3. Hierarchical Structure

- The hierarchy of Animals, Dogs, and Cats was designed using a one-to-one relationship, where Dogs and Cats are subcategories of Animals.
- This allows the system to maintain a single source of truth for general animal attributes while supporting species-specific details.

4. Scalability

- The design accommodates future expansions. For example, adding more species (e.g., Birds or Rabbits) would involve creating additional tables that link back to the Animals table.

5. Challenges

- Maintaining the integrity of hierarchical relationships between general (Animals) and specific entities (Dogs and Cats) required ensuring the proper use of primary and foreign keys.
- Designing AdoptionRequests to link to multiple entities (Animals, Adopters, and ShelterEmployees) required careful mapping to prevent circular dependencies.

6. Practical Use

- Data types were chosen to match expected input, such as VARCHAR for descriptive attributes and DATE for timestamps.

- Each table was designed to include only relevant fields, avoiding unnecessary duplication while providing sufficient detail for efficient queries.

This design ensures that the Animal Shelter Management System is robust, flexible, and efficient, supporting both current operational needs and potential future growth. Let me know if further refinements or additions are needed!