# PET ADOPTION

## Database Management System

UNDER SUPERVISION
DR. WALAA MEDHAT
ENG. NADA SALAH

THE TEAM
Malak Ahmed 221001692
Haya Osama 221001798
Doaa Mahdy 221000424
Farida Ashraf 221000407
Farida Hazem 221000214

**Project Title: Pet Adoption and Management System**

# ➢ Project Description:

The **Pet Adoption and Management System** is a database-driven application designed to streamline the operations of animal shelters and facilitate pet adoptions. This system ensures efficient management of shelter operations, animal care, and adoption processes by maintaining detailed records and enabling effective follow-ups.

The database includes modules to:

1. Manage shelter information, capacity, and locations.
2. Keep track of animals, including species, breed, health status, and availability for adoption.
3. Record adopters' details and their adoption status, including historical records.
4. Document animals' medical histories to ensure proper healthcare management.
5. Handle employee data and hierarchical relationships, including supervision details.
6. Manage adoption follow-ups to monitor post-adoption animal welfare.

# ➢ Project Objective :

The **Pet Adoption Management System** streamlines shelter operations by managing pet adoption processes, medical histories, and employee assignments through a robust database-driven platform. It ensures data accuracy, minimizes redundancy, and provides user-friendly tools for efficient data entry, retrieval, and reporting.

---

# ➢ Business Rules for the Pet Adoption System Database

**1. Shelter Management**

1. Each shelter in the system must have a unique (shelterID) to distinguish it from other shelters.
2. A shelter must have a name, location, and valid contact information.
    - **Attributes:** shelterName, shelterLocation, shelterContactInfo
3. Each shelter has a maximum capacity (shelterCapacity), which must not be exceeded when assigning animals.
4. Animals housed in a shelter must be linked via (shelterID).

---

**2. Animal Management**

1. Every animal in the database must have a unique (animalID).
2. Each animal record must include:
    - **Attributes:** animalName, animalSpecies, animalBreed, animalAge
3. The health status (animal_Health_Status) of an animal must be recorded and updated as needed to reflect its current condition.
4. The availability status (animalAvailabilty) must be:
    - TRUE if the animal is available for adoption.
    - FALSE if the animal is already adopted or not adoptable.
5. Each animal must belong to a specific shelter (shelterID) as a Foreign Key and cannot exist without a shelter assignment.

---

**3. Adopter Management**

1. Each adopter must have a unique (adopterID) to identify them.
2. Adopters must provide:
    - Personal details: adopterName, adopterAdress.
    - Contact information: adopterEmail, adopterPhone_Number.

3. The application status (applicationStatus) of type Enum of an adopter must be one of:
    - Rejected
    - In Progress
    - Accepted
4. An adoption date (adoptionDate) is only recorded if the (applicationStatus) is Accepted.
5. Each adopter may be linked to a specific animal (animalID) and shelter (shelterID) upon adoption (Foreign Keys).

---

**4. Medical History**

1. Each medical history entry must have a unique (medID).
2. The medical history must record the treatment name (treatmentName) and the date of the treatment (treatmentDate).
3. Each medical history record must link to an existing animal via its (animalID) (Foreign Key).

---

**5. Employee Management**

1. Each employee must have a unique (employeeID) to identify them.
2. Employees must provide:
    - Personal details: employeeName, employeeEmail.
    - Employment details: employeeSalary.
3. Employees may optionally have a supervisor (supervisorID) which is Self-Referential Foreign Key from (employeeID)who must also be an employee in the system.

---

**6. Adoption Follow-Up**

1. Each adoption follow-up is uniquely identified by a Composite Primary Key of (employeeID) and (adopterID) Which are Foreign Keys.
2. Follow-ups must include:
    - The date of the follow-up (followUpDate).
    - Notes describing the follow-up (notes).
3. Each follow-up must be conducted by a valid employee (employeeID) which is a Foreign Key.
4. Each follow-up must be linked to an existing adopter (adopterID) which is a Foreign Key.

---

# ➢ General Business Rules

1. **Data Integrity:**
    - All foreign key constraints must be enforced to ensure relationships are valid.
2. **Deletion Rules:**

    **If a shelter is deleted**:

    - All associated animals must either:
        - **Be reassigned to another shelter**: The shelterID in the Animal table is set to NULL, allowing manual reassignment later.
        - **Or deleted**: This is optional and can be configured using a cascade delete if desired. Currently, the system will set shelterID to NULL.

    **If an animal is deleted**:

    - All linked medical history records will be **automatically deleted** to maintain data consistency.

- Any linked adoption records will also be **automatically deleted**, as they are directly tied to the animal.

 **If an adopter is deleted**:

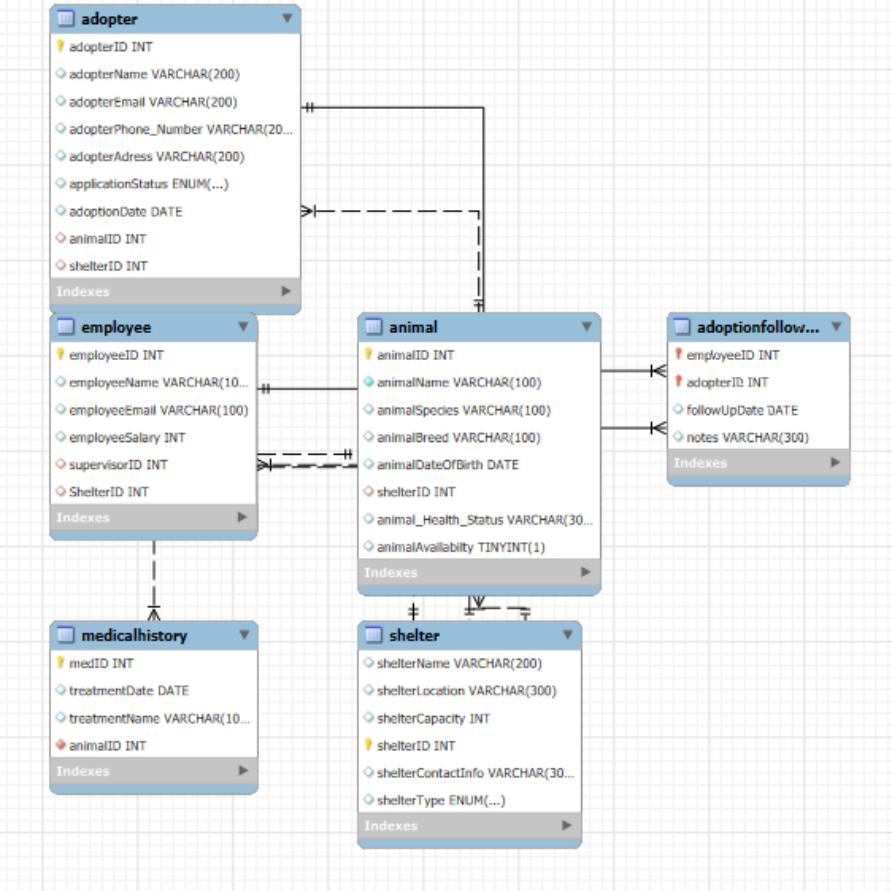
- All associated adoption follow-up records will be **automatically deleted**, as they become irrelevant without the adopter.

 **If an employee is deleted**:

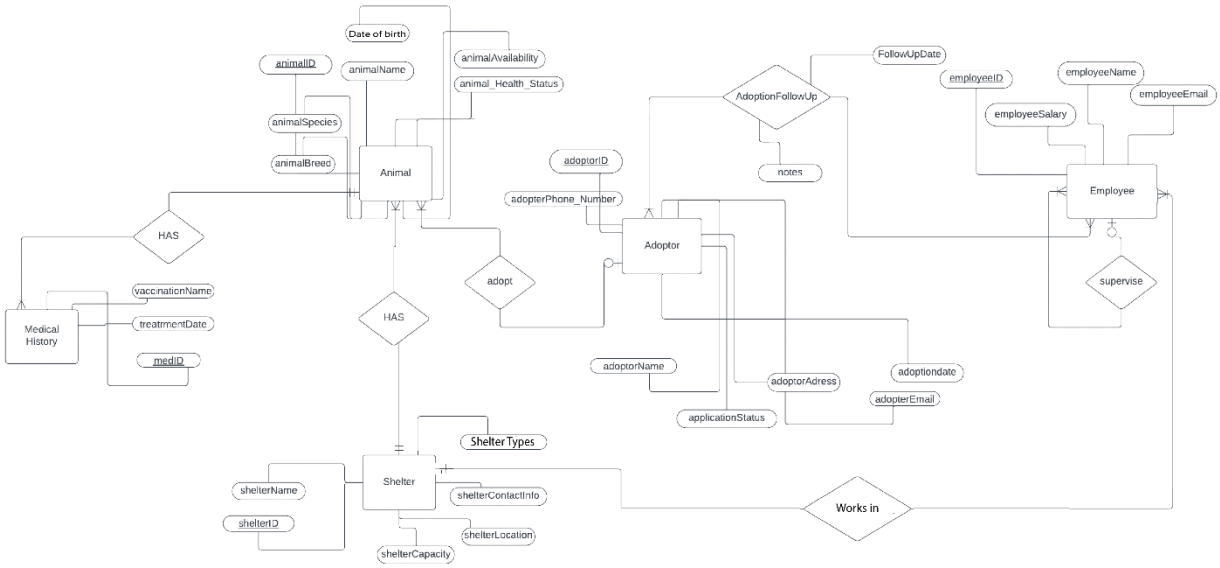
- All linked adoption follow-up records must either:
  - **Be reassigned to another employee**: This requires setting the employeeID in the AdoptionFollowUp table to NULL, allowing for manual reassignment.
  - **Or deleted**: If reassignment is not possible or required, the follow-up records will be removed.

3. **Auditing and Reporting:**
   - Reports should summarize:
     - Animal health trends from medical history.
     - Adoption statistics, including success rates and follow-up actions.
     - Employee performance in conducting follow-ups.

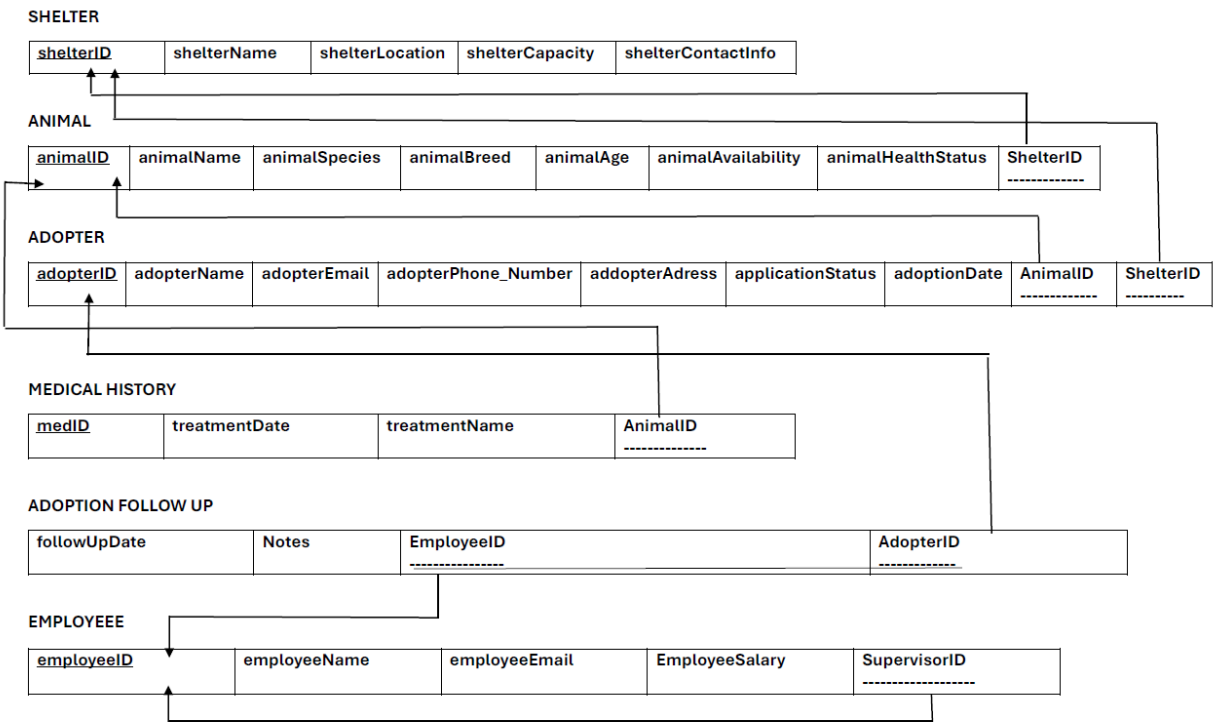
# ➢ ERD Diagram:




## ➢ ERD Chart:




## ➢ Mapping :




# ➢ Functional dependencies for each table:

- Shelter Table:
  Attributes : ShelterID, ShelterName, shelterLocation, shelterCapacity, shelterContactInfo, shelterType
  - Dependencies:

  ShelterID → ShelterName, Shelter Location ,sheleterCapicity, ShelterContactInfo,ShelterType
  ShelterName → ShelterID
- Animal Table:
  Attributes: animalID, animalName, animalSpecies, animalBreed, animalDateOfBirth, shelterID, animal_Health_Status, animalAvailability
  - Dependencies:

  animalID → animalName, animalSpecies, animalBreed, animalDateOfBirth, shelterID, animal_Health_Status, animalAvailability
  shelterID → shelterName
- Adopter Table:

Attributes: adopterID, adopterName, adopterEmail, adopterPhone_Number, adopterAddress, applicationStatus, adoptionDate, animalID

- Dependencies:

adopterID →adopterName, adopterEmail, adopterPhone_Number, adopterAddress, applicationStatus, adoptionDate, animalID
adopterEmail → adopterID
animalID → animalName

- MedicalHistory Table:
  Attributes: medID, treatmentDate, treatmentName, animalID
  - Dependencies:

  medID → treatmentDate, treatmentName, animalID
  animalID →animalName
- AdoptionFollowUp Table:
  Attributes: employeeID, adopterID, followUpDate, notes
  - Dependencies:

  (employeeID, adopterID) → followUpDate, notes
  employeeID → employeeName
  adopterID → adopterName
- Employee Table:
  Attributes: employeeID, employeeName, employeeEmail, employeeSalary
  - Dependencies:

  employeeID → employeeName, employeeEmail, employeeSalary
  employeeEmail → employeeID
- EmployeeShelter Table:
  Attributes: employeeID, shelterID
  - Dependencies:

  (employeeID, shelterID) → employeeName, shelterName
- AnimalShelter Table :
  Attributes: animalID, shelterID
  - Dependencies:

  (animalID,ShelterID) → animalName,ShelterName

- SupervisorAssignment Table:
  Attributes: employeeID, supervisorID
    - Dependencies:
- employeeID → supervisorID

# ➢ Normalization :