ВАРНЕНСКИ
СВОБОДЕН
УНИВЕРСИТЕТ
ЧЕРНОРИЗЕЦ ХРАБЪР

Faculty of International Economics and Administration

Computer Science Department

Course Project

Databases

Final Project

Relational database

Online animal adoption and donation system

Submitted by: Elis Mehmed

FNº 213110003

2023

Table of Contents

Description of the subject area

The subject area of this project is the development of a web application that aims to automate the process of adoption of animals and make easy donations for animals and shelters. The application will provide a user-friendly interface that allows users to search animals in defferent shelters in different locations and view their profiles.

Designing the structure of the database

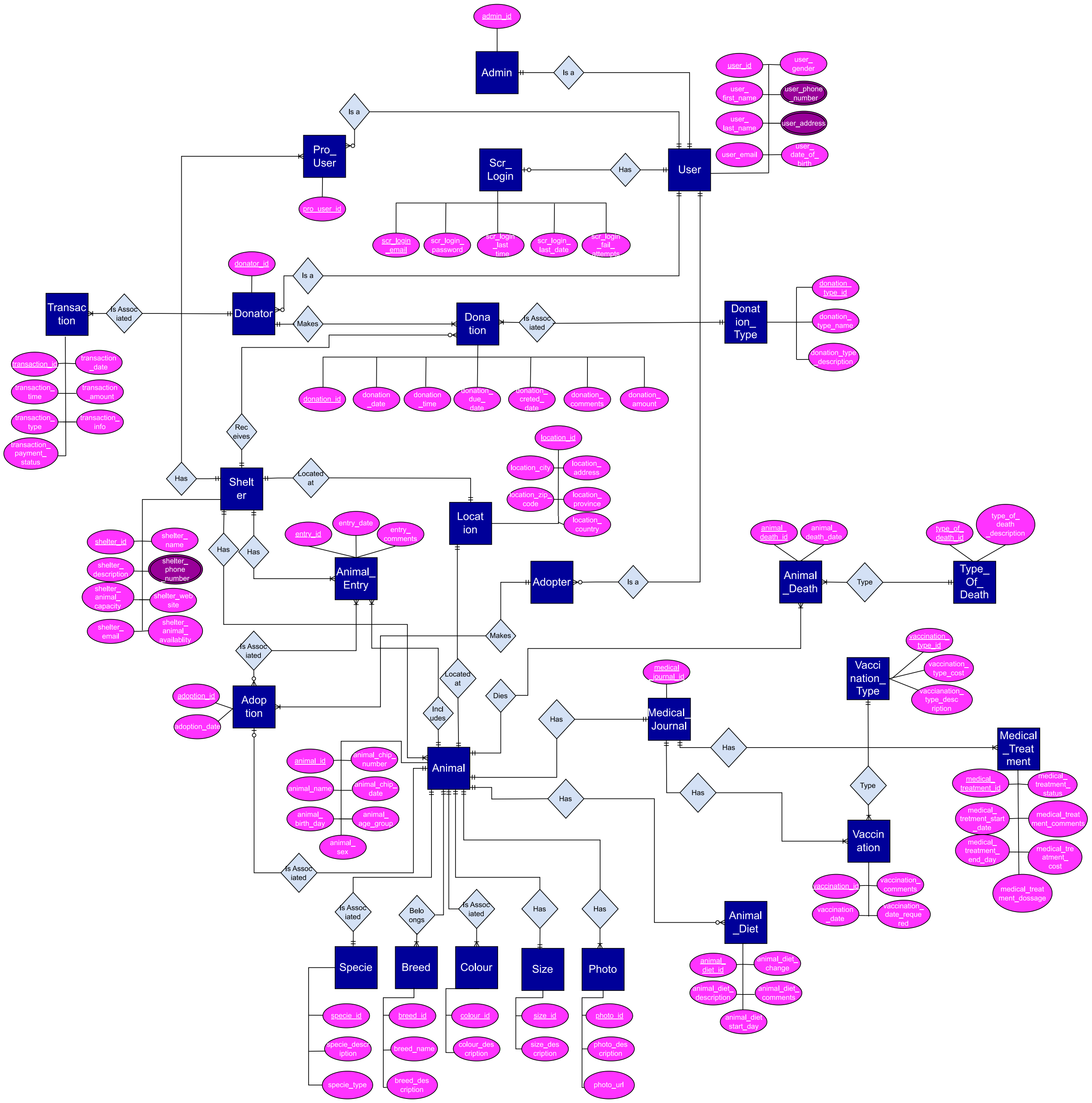
The E-R model for this database will include the following entities: Scr_Login, Admin, User, Pro_User, Shelter, Adopter, Adoption, Animal, Breed, Size, Specie, Colour, Photo, Medical_Journal, Death, Medical_Treatment, Animal_Diet, Transaction, Donation, Donation_type, Entry, Location, Type_Of_Death, Vaccination, Vaccination_Type , Transaction.

The relationships between these entities are as follows:

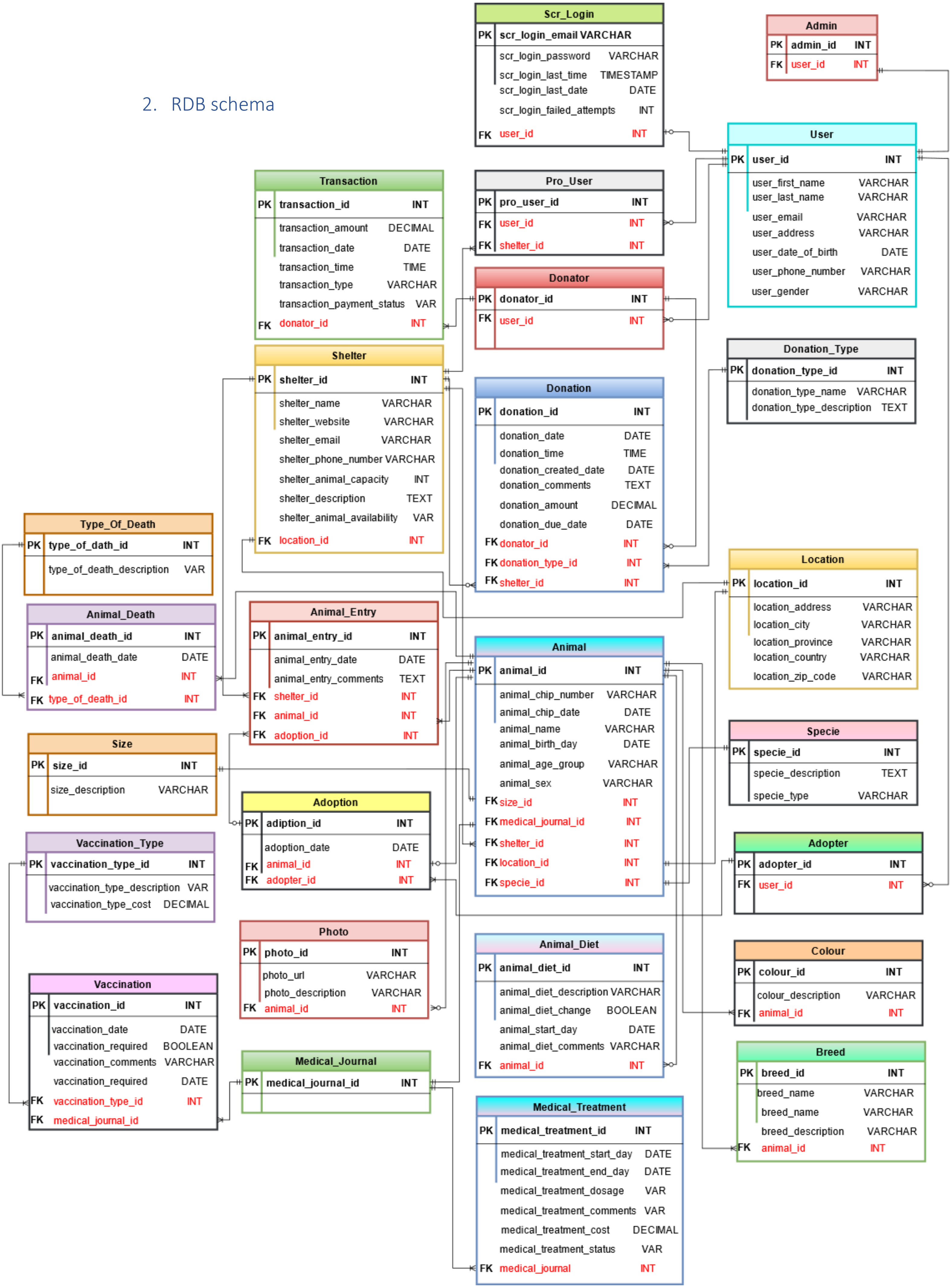
- An Admin is a User.
- An Admin can have zero or one SecureLogin.
- Each User can have zero or one SecureLogin.
- Each ProUser can have zero or one SecureLogin.
- Each Shelter has one or more ProUser.
- For each Donator corresponds exectly one User.
- Each Donation is associated to exactly one TypeOfDonation.
- Shelters can receive zero or more donations from Donators.
- One Donator can have one or more Transactions.
- Each Shelter is located at exactly one Location.
- Each Shelter can have one or more AnimalEntry.

- It can be zero or more Adoptions that is Associate with the AnimalEntry.
- Each Animal can have one or more AnimalEntry if it si returned after the Adoption for some reasons back to the Shelter.
- Each Shelter has one or more Animlas.
- AnimalDeath Entity has the records for ecach death Animal.
- Each AnimalDeath is of axactly one TypeOfDeath.
- Each Adopter corresponds to exactly one User.
- Each Animal is located at exactly one Location.
- Each Animal can have one or more Breed.
- Ecah Animal belongs to exactly one Specie.
- Each Animal can have one or more Colours.
- Each Animal has a size.
- Each Animal can have one or more photos.
- Each Animal has an exactly one Medical_Journal.
- Each MedicalJournal can have one or more Vaccinations.
- Each Vaccination corresponds to exactly one TypeOfVaccination.
- Each MedicalJournal can have one or more MedicalTreatments.

1. E-R model for the online animal adoption and donation system



2. RDB schema



	Scr_Login	
PK	scr_login_email VARCHAR	
	scr_login_password	VARCHAR
	scr_login_last_time	TIMESTAMP
	scr_login_last_date	DATE
	scr_login_failed_attempts	INT
FK	user_id	INT

	Admin	
PK	admin_id	INT
FK	user_id	INT

	User	
PK	user_id	INT
	user_first_name	VARCHAR
	user_last_name	VARCHAR
	user_email	VARCHAR
	user_address	VARCHAR
	user_date_of_birth	DATE
	user_phone_number	VARCHAR
	user_gender	VARCHAR

	Transaction	
PK	transaction_id	INT
	transaction_amount	DECIMAL
	transaction_date	DATE
	transaction_time	TIME
	transaction_type	VARCHAR
	transaction_payment_status	VAR
FK	donator_id	INT

	Pro_User	
PK	pro_user_id	INT
FK	user_id	INT
FK	shelter_id	INT

	Donator	
PK	donator_id	INT
FK	user_id	INT

	Donation_Type	
PK	donation_type_id	INT
	donation_type_name	VARCHAR
	donation_type_description	TEXT

	Shelter	
PK	shelter_id	INT
	shelter_name	VARCHAR
	shelter_website	VARCHAR
	shelter_email	VARCHAR
	shelter_phone_number	VARCHAR
	shelter_animal_capacity	INT
	shelter_description	TEXT
	shelter_animal_availability	VAR
FK	location_id	INT

	Donation	
PK	donation_id	INT
	donation_date	DATE
	donation_time	TIME
	donation_created_date	DATE
	donation_comments	TEXT
	donation_amount	DECIMAL
	donation_due_date	DATE
FK	donator_id	INT
FK	donation_type_id	INT
FK	shelter_id	INT

	Type_Of_Death	
PK	type_of_dath_id	INT
	type_of_death_description	VAR

	Location	
PK	location_id	INT
	location_address	VARCHAR
	location_city	VARCHAR
	location_province	VARCHAR
	location_country	VARCHAR
	location_zip_code	VARCHAR

	Animal_Death	
PK	animal_death_id	INT
	animal_death_date	DATE
FK	animal_id	INT
FK	type_of_death_id	INT

	Animal_Entry	
PK	animal_entry_id	INT
	animal_entry_date	DATE
	animal_entry_comments	TEXT
FK	shelter_id	INT
FK	animal_id	INT
FK	adoption_id	INT

	Animal	
PK	animal_id	INT
	animal_chip_number	VARCHAR
	animal_chip_date	DATE
	animal_name	VARCHAR
	animal_birth_day	DATE
	animal_age_group	VARCHAR
	animal_sex	VARCHAR
FK	size_id	INT
FK	medical_journal_id	INT
FK	shelter_id	INT
FK	location_id	INT
FK	specie_id	INT

	Size	
PK	size_id	INT
	size_description	VARCHAR

	Specie	
PK	specie_id	INT
	specie_description	TEXT
	specie_type	VARCHAR

	Adoption	
PK	adiption_id	INT
	adoption_date	DATE
FK	animal_id	INT
FK	adopter_id	INT

	Vaccination_Type	
PK	vaccination_type_id	INT
	vaccination_type_description	VAR
	vaccination_type_cost	DECIMAL

	Adopter	
PK	adopter_id	INT
FK	user_id	INT

	Photo	
PK	photo_id	INT
	photo_url	VARCHAR
	photo_description	VARCHAR
FK	animal_id	INT

	Colour	
PK	colour_id	INT
	colour_description	VARCHAR
FK	animal_id	INT

	Animal_Diet	
PK	animal_diet_id	INT
	animal_diet_description	VARCHAR
	animal_diet_change	BOOLEAN
	animal_start_day	DATE
	animal_diet_comments	VARCHAR
FK	animal_id	INT

	Vaccination	
PK	vaccination_id	INT
	vaccination_date	DATE
	vaccination_required	BOOLEAN
	vaccination_comments	VARCHAR
	vaccination_required	DATE
FK	vaccination_type_id	INT
FK	medical_journal_id	

	Medical_Journal	
PK	medical_journal_id	INT

	Breed	
PK	breed_id	INT
	breed_name	VARCHAR
	breed_name	VARCHAR
	breed_description	VARCHAR
FK	animal_id	INT

	Medical_Treatment	
PK	medical_treatment_id	INT
	medical_treatment_start_day	DATE
	medical_treatment_end_day	DATE
	medical_treatment_dosage	VAR
	medical_treatment_comments	VAR
	medical_treatment_cost	DECIMAL
	medical_treatment_status	VAR
FK	medical_journal	INT

Description of the table structure

- **Admin table:**
    - admin_id: primary key. This field uniquely identifies each administrator in the table.
    - user_id: foreign key. This field indicates the relationship between Admin and User.

- **Scr_Login table:**
    - scr_login_email: primary key. This field uniquely identifies each login account.
    - scr_login_password: password associated with the login account. The field will store the encrypted version of the password for security reasons.
    - scr_login_last_time: the last time of successful login for the account. The field will store the timestamp of the last successful login and will be updated every time a user successfully login.
    - scr_login_last_date: The field will store the date of the last successful login.
    - scr_login_last_attempts: The field will store the number of consequitive failed login attempts for the account and will be rest to zero when the user succesfully logs in. The field can be used to implement security measures, such as temporarily locking the account after certain number of failed login attempts.
    - user_id: foreign key. It is used to link a user's login credentials to their corresponding user record in the User table.

- **User table:**
  - user_id: primary key. This field uniquely identifies each user in the system.
  - first_name: the first name of the user.
  - last_name: the last name of the user.
  - gender: the gender of the user.
  - phone_number: The field will store one or more phone numbers of the user. This will be implemented as a comma-separated list of phone numbers in a single field.
  - address: The field will store one or more addresse of the user. This will be implemented as a comma-separated list of addresses in a single field.
  - date_of_birth: the date of birth of the user.
  - email: the email address of the user.

- **Pro_User table:**
  - pro_user_id: primary key. This field uniquely identifies each pro user in the table.
  - user_id: foreign key: This field links each Pro_User record to a specific user record in the User table.
  - shelter_id: foregin key: This field links each Pro_User record to a specific record in the Sheleter.

- **Donator table:**
  - donator_id: primary key. This field uniquely identifies each donator in the table.
  - user_id: foregin key: This field links each Donator record to a specific record in the User table.

- **Donation table:**
  - donation_id: primary key. This field uniquely identifies each donation in the table.
  - donation_date: the date of the donation.
  - donation_time: the time of the donation.
  - donation_created_day: the day the donation was created in the system.
  - donation_comments: any comments associated with the donation.
  - donation_amount: the amount of the donation.
  - donation_due_date: the due day of the donation (if applicable).
  - donator_id: foreign key. Represents the person who did the donation.
  - donation_type_id: foreign key. The type of donation that was made.
  - shelter_id: foreign key. The shelter where the donation was made for.

- **Donation_Type table:**
  - donation_type_id: primary key. This field uniquely identifies each donation type in the table.
  - donation_type_name: the name of the donation type.
  - donation_type_description: a description of the donation type.
- **Transaction table:**
  - transaction_id: primary key. This field uniquely identifies each transaction in the table.
  - transaction_date: the date of the transaction.
  - transaction_time: the time of the transaction.
  - transaction_amount: the amount of the transaction.

  - transaction_info: any additional information associated with the transaction.
  - transaction_type: the type of the transaction (e.g. refund, payment).
  - transaction_payment_status: the status of the payment assosiated with the transaction (e.g. paid, pending, declined).
  - donator_id: foreign key. This field will allow the system to associate each transaction with a particular donator, and to track the total amount by each donor over time.

- **Shelter table:**
  - shelter_id: primary key. This field uniquely identifies each shelter in the table
  - shelter_name: name of the shelter.
  - shelter_phone number: The field will store one or more phone numbers of the shelter. This will be implemented as a comma-separated list of phone numbers in a single field.
  - shelter_website: the website for the shelter
  - shelter_animal_availability: information about the types of animals available at the shelter.
  - shelter_email: the email address of the shelter
  - shelter_animal_capacity: the maximum number of animals that the shelter can accommodate.
  - shelter_id: description: the description of the shelter.
  - location_id: foreign key: This field associates each shelter with a specific location and facilitites searches for shelters in a particular geographic area.

- **Adopter table:**
    - adopter_id: primary key. This field uniquely identifies each adopter in the table.
    - user_id: foreign key. This field links each Adopter record to a specific user record in the User table.

- **Adoption table:**
    - adoption_id: primary key. This field uniquely identifies each adoption in the table.
    - adoption_date: the date of the adoption.
    - animal_id: foreign key. This field indicates which animal was adopted.
    - adopter_id: foregn key. This field indicates which user adopted the animal.

- **Location table:**
    - location_id: primary key. This field uniquely identifies each location in the table.
    - location_country: the name of the country where is the location.
    - location_zip_code: the postal code of the location.
    - location_address: The field will store the exact address of the animal and shelter. This will be implemented as a comma-separated list of addresses in a single field.
    - location_province: the name of the province where is the location.
    - location_city: the name of the city where is the location.

- **Animal_Entry table:**
    - animal_entry_id: primary key. This field uniquely identifies each animal entry in the table.

- animal_entry_date: the date animal entered in the shelter.
- animal_entry_comments: any additional comments associated with the animal entry.
- shelter_id: foreign key. This field references the shelter that received the animal entry.
- animal_id: foreign key. The animal that was entered into the shelter.
- adoption_id: foreign key. In case of adoption the Animal entry will link with Adoption table to adjust in and out animal balance.

- **Animal table:**
  - animal_id: primary key. This field uniquely identifies each animal in the shelter.
  - animal_name: the name of the animal(if applicable).
  - animal_chip_number: the unique identification number associated with the animal's microchip.
  - animal_chip_date: the date the chip was inserted.
  - animal_birth_day: the date of the animal's birth.
  - animal_age_group: a grouping of animals based on their age (puppy, adult, senior).
  - animal_sex: the sex of the animal (e.g. male, female, unknown).
  - **size_id: foreign key:** This field associates each animal with a specific size.
  - **shelter_id: foreign key:** This field associates each animal with the specific shelter.
  - **medical_journal_id: foreign key:** Each animal is assosiated to one medical journal.
  - **location_id: foreign key:** Each animal is assosiated to one location.

  - **specie_id: foreign key:** Each animal is assosiated with exactly one specie.

- **Specie table:**
  - specie_id: primary key. This field uniquely identifies each specie in the table.
  - specie_type: the animal type to which the species belongs (e.g. dog, cat, bird).
  - specie_description: a brief description of the species.

- **Breed table:**
  - breed_id: primary key. This field uniquely identifies each breed in the table.
  - breed_name: the name of the breed.
  - breed_description: a brief description of the breed.
  - animal_id: foreign key. Which animal belongs to each breed.

- **Size table:**
  - size_id: primary key. This field uniquely identifies each size in the table.
  - size_description: a brief description of the size (small, medium, large).

- **Colour table:**
  - colour_id: primary key. This field uniquely identifies different animal colour in the table.
  - colour_description: a brief description of the colour (black, brown, white).

  - animal_id: foreign key. Associating a particular colour with a specific animal.

- **Photo table:**
  - photo_id: primary key. This field uniquely identifies each photo in the table.
  - photo_description: a brief description of the photo.
  - photo_url: the URL or file path for the photo.
  - animal_id: foreign key: This field allows to link photos with the corresponding animal in the database.

- **Animal_Diet table:**
  - animal_diet_id: primary key. This field uniquely identifies each animal diet in the table.
  - animal_diet_description: a brief description of the diet plan.
  - animal_diet_change: a Boolen field indicating wheather the diet plan has changed recently.
  - animal_diet_start_day: the date when the diet plan has started.
  - animal_diet_comments: any additional comments for the diet plan.
  - animal_id: foreign key. This establish relation betweeen animal and its diet.

- **Animal_Death table:**
  - animal_death_id: primary key. This field uniquely identifies each animal death in the table.

  - animal_death_date: the date when the animal passed away.
  - animal_id: foreign key. This field assosiates the animal that has died.
  - type_of_death. This field assosiates the type of death the animl has had.

- **Type_Of_Death table:**
  - type_of_death_id: primary key. This field uniquely identifies each type of death in the table.
  - type_of_death_description: a brief description of the type of death.

- **Medical_Journal table:**
  - Medical_journal_id: primary key. This field uniquely identifies each medical journal in the table.

- **Medical_Treatment table:**
  - medical_treatment_id: primary key. This field uniquely identifies each medical treatment in the table.
  - medical_treatment_start_day: the date when the medical treatment has started.
  - medical_treatment_end_day: the date when the medical treatment has ended.
  - medical_treatment_comments: any additional comments for the medical tretment.
  - medical_treatment_cost: the cost of the treatment.
  - medical_treatment_dossage: the dossage of the treatment.
  - medical_treatment_status: the staus of the treatment (completed, ongoing, cancelled).

  - medical_journal_id: foreign key. Each medical treatment record can be associated with a specific medical journal.

- **Vaccination table:**
  - vaccination_id: primary key. This field uniquely identifies each vaccination in the table.
  - vaccination_date: the date when the vaccination was given.
  - vaccination_requered: A Boolen field indicating if the vaccination is required or not.
  - vaccination_comments: any additional comments for the vaccination.
  - vaccination_type_id: foreign key. Indicates the vaccination type was addministrated.
  - medical_journal_id: foreign key. Indicates the vaccines recorded in medical journal of the animal.

- **Vaccination_Type table:**
  - Vaccination_type_id: primary key. This field uniquely identifies each vaccination type in the table.
  - vaccination_type_description: a brief decription of the vaccination type.
  - vaccination_type_cost: the cost of the vaccination type.

DDL script to create the online animal adoption and donation system.

The following is a DDL script to create the online animal adoption and donation system.

```
CREATE TABLE Admin (
   admin_id INT PRIMARY KEY,
   user_id INT,
   FOREIGN KEY (user_id) REFERENCES User(user_id)
);

CREATE TABLE Scr_Login (
  scr_login_email VARCHAR (255) PRIMARY KEY,
  scr_login_password VARCHAR (255) NOT NULL,
  scr_login_last_time TIMESTAMP,
  scr_login_last_date DATE,
  scr_login_last_attempts INT DEFAULT 0,
  user_id INT,
  FOREIGN KEY (user_id) REFERENCES User(user_id)
);

CREATE TABLE User (
  user_id INT PRIMARY KEY,
  user_ first_name VARCHAR (255),
  user_last_name VARCHAR (255),
  user_gender VARCHAR (10),
  user_phone_number VARCHAR (255),
  user_address VARCHAR (255),
  user_date_of_birth DATE,
  user_email VARCHAR (255)
);

CREATE TABLE Pro_User (
  pro_user_id INT PRIMARY KEY,
  user_id INT,
  shelter_id INT,
  FOREIGN KEY (user_id) REFERENCES User(user_id),
  FOREIGN KEY (shelter_id) REFERENCES Shelter(shelter_id)
);
```

```
CREATE TABLE Donator (
  donator_id INT PRIMARY KEY,
  user_id INT NOT NULL,
  FOREIGN KEY (user_id) REFERENCES User(user_id)
);

CREATE TABLE Donation (
  donation_id INT PRIMARY KEY,
  donation_date DATE,
  donation_time TIME,
  donation_created_day DATE,
  donation_comments TEXT,
  donation_amount DECIMAL(10,2),
  donation_due_date DATE,
  donator_id INT,
  donation_type_id INT,
  shelter_id INT,
  FOREIGN KEY (donator_id) REFERENCES Donator(donator_id),
  FOREIGN KEY (donation_type_id) REFERENCES Donation_Type(donation_type_id),
  FOREIGN KEY (shelter_id) REFERENCES Shelter(shelter_id)
);

CREATE TABLE Donation_Type (
  donation_type_id INT PRIMARY KEY,
  donation_type_name VARCHAR(50),
  donation_type_description TEXT
);

CREATE TABLE Transaction (
    transaction_id INT PRIMARY KEY,
    transaction_date DATE NOT NULL,
    transaction_time TIME NOT NULL,
    transaction_amount DECIMAL(10,2) NOT NULL,
    transaction_info VARCHAR(255),
    transaction_type VARCHAR(20) NOT NULL,
    transaction_payment_status VARCHAR(20) NOT NULL,
    donator_id INT,
    FOREIGN KEY (donator_id) REFERENCES Donator(donator_id)
);

CREATE TABLE Shelter (
  shelter_id INT PRIMARY KEY,
  shelter_name VARCHAR(255) NOT NULL,
  shelter_phone_number VARCHAR(255),
  shelter_website VARCHAR(255),
  shelter_animal_availability VARCHAR(255),
  shelter_email VARCHAR(255),
  shelter_animal_capacity INT,
  shelter_description TEXT,
  location_id INT,
  FOREIGN KEY (location_id) REFERENCES Location(location_id)
);

CREATE TABLE Adopter (
  adopter_id INT PRIMARY KEY,
  user_id INT,
  FOREIGN KEY (user_id) REFERENCES User(user_id)
);

CREATE TABLE Adoption (
  adoption_id INT PRIMARY KEY,
  adoption_date DATE,
  animal_id INT,
  adopter_id INT,
  FOREIGN KEY (animal_id) REFERENCES Animal(animal_id),
  FOREIGN KEY (adopter_id) REFERENCES Adopter(adopter_id)
);

CREATE TABLE Location (
  location_id INT PRIMARY KEY,
  location_country VARCHAR(50) NOT NULL,
  location_zip_code VARCHAR(10) NOT NULL,
  location_address VARCHAR(255) NOT NULL,
  location_province VARCHAR(50) NOT NULL,
  location_city VARCHAR(50) NOT NULL
);

CREATE TABLE Animal_Entry (
  animal_entry_id INT PRIMARY KEY,
  animal_entry_date DATE NOT NULL,
  animal_entry_comments TEXT,
  shelter_id INT NOT NULL,
  animal_id INT NOT NULL,
  adoption_id INT,
  FOREIGN KEY (shelter_id) REFERENCES Shelter(shelter_id),
  FOREIGN KEY (animal_id) REFERENCES Animal(animal_id),
  FOREIGN KEY (adoption_id) REFERENCES Adoption(adoption_id)
);
```

```
CREATE TABLE Animal (
    animal_id INT PRIMARY KEY,
    animal_name VARCHAR(50),
    animal_chip_number VARCHAR(20) UNIQUE,
    animal_chip_date DATE,
    animal_birth_day DATE,
    animal_age_group VARCHAR(10),
    animal_sex VARCHAR(10),
    size_id INT,
    shelter_id INT,
    medical_journal_id INT,
    FOREIGN KEY (size_id) REFERENCES Size(size_id),
    FOREIGN KEY (shelter_id) REFERENCES Shelter(shelter_id),
    FOREIGN KEY (medical_journal_id) REFERENCES MedicalJournal(medical_journal_id)
    FOREIGN KEY (location_id) REFERENCES Location(location_id),
    FOREIGN KEY (lspecie_id) REFERENCES Specie(specie_id),
);

 CREATE TABLE Specie (
  specie_id INT PRIMARY KEY,
  specie_type VARCHAR(50),
  specie_description VARCHAR(255)
);

CREATE TABLE Breed (
  breed_id INT PRIMARY KEY,
  breed_name VARCHAR(50),
  breed_description VARCHAR(255),
  animal_id INT,
  FOREIGN KEY (animal_id) REFERENCES Animal (animal_id)
);

CREATE TABLE Size (
  size_id INT PRIMARY KEY,
  size_description VARCHAR(50)
);

CREATE TABLE Colour (
  colour_id INT PRIMARY KEY,
  colour_description VARCHAR(50),
  animal_id INT,
  FOREIGN KEY (animal_id) REFERENCES Animal (animal_id)
);
```

```
 CREATE TABLE Photo (
  photo_id INT PRIMARY KEY,
  photo_description VARCHAR(255),
  photo_url VARCHAR(255),
  animal_id INT,
  FOREIGN KEY (animal_id) REFERENCES Animal (animal_id)
);

CREATE TABLE Animal_Diet (
  animal_diet_id INT PRIMARY KEY,
  animal_diet_description VARCHAR(255),
  animal_diet_change BOOLEAN,
  animal_diet_start_day DATE,
  animal_diet_comments VARCHAR(255),
  animal_id INT,
  FOREIGN KEY (animal_id) REFERENCES Animal(animal_id)
);

CREATE TABLE Animal_Death (
  animal_death_id INT PRIMARY KEY,
  animal_death_date DATE,
  animal_id INT,
  type_of_death VARCHAR(255),
  FOREIGN KEY (animal_id) REFERENCES Animal(animal_id)
);

CREATE TABLE Type_Of_Death (
  type_of_death_id INT PRIMARY KEY,
  type_of_death_description VARCHAR(255)
);

CREATE TABLE Medical_Journal (
  Medical_journal_id INT PRIMARY KEY,
);


CREATE TABLE Medical_Treatment (
  medical_treatment_id INT PRIMARY KEY,
  medical_treatment_start_day DATE,
  medical_treatment_end_day DATE,
  medical_treatment_comments VARCHAR(255),
  medical_treatment_cost DECIMAL(10,2),
  medical_treatment_dossage VARCHAR(255),
  medical_treatment_status VARCHAR(255),
  medical_journal_id INT,
  FOREIGN KEY (medical_journal_id) REFERENCES Medical_Journal(Medical_journal_id)
);


CREATE TABLE Vaccination (
  vaccination_id INT PRIMARY KEY,
  vaccination_date DATE,
  vaccination_required BOOLEAN,
  vaccination_comments VARCHAR(255),
  vaccination_type_id INT,
  animal_id INT,
  FOREIGN KEY (vaccination_type_id) REFERENCES Vaccination_Type(vaccination_type_id),
  FOREIGN KEY (medical_journal) REFERENCES Medical_journal(medical_journal_id)
);

CREATE TABLE Vaccination_Type (
  Vaccination_type_id INT PRIMARY KEY,
  vaccination_type_description VARCHAR(255),
  vaccination_type_cost DECIMAL(10,2)
);
```