

**Week 2 Project Report**  
**Database (MII212501)**



**PawPoint: Veterinary Clinic Appointment Scheduling and Management  
Application**

**Written By:**

**Deira Aisya Refani (24/532821/PA/22539)**

**Ratu Faiha Salsabilla (24/532756/PA/22533)**

**Aufaa Azzahra Aryawan (24/546854/PA/23215)**

**Universitas Gadjah Mada**

**Faculty Of Mathematics and Natural Science**

**Department of Computer Science and Electronics**

**2025**

## 1. Overview of the SQL Script

The SQL script defines the core database structure for the veterinary clinic system. It includes the following tables:

- User – stores all user accounts (superadmin, admin, veterinarian, and pet owner)
- PetOwner – stores the pet owner details linked to user accounts
- Veterinarian – stores veterinarian details linked to user accounts
- Clinic – stores clinic details
- Pet -
- Appointment -
- TreatmentRecord -

## 2. Explanation of Tables and Attributes

### A. User table

```
CREATE TABLE User (  
  user_id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(50) NOT NULL UNIQUE,  
  password VARCHAR(300) NOT NULL,  
  role ENUM('superadmin', 'admin', 'vet', 'pet owner') NOT NULL,  
  email VARCHAR(100),  
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

Each user is uniquely identified by *user\_id*. The *username* attributes are 'unique' and 'NOT NULL' and is to ensure that every user has a distinct username and to prevent duplicate logins - maintaining the integrity of the authentication system. Whilst, *password* is NOT NULL for authentication of the user. Moreover, the *role* column uses an 'ENUM' to restrict values to validated user types (superadmin, admin, vet, and pet\_owner). Lastly, *email* is optional for contact and *created\_at* defaults to the current timestamp to record the creation of the account.

### B. PetOwner table

```
CREATE TABLE PetOwner (  
  owner_id INT AUTO_INCREMENT PRIMARY KEY,  
  user_id INT NOT NULL,  
  name VARCHAR(100) NOT NULL,  
  phone VARCHAR(20),  
  address VARCHAR(300),  
  FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE RESTRICT,  
  CHECK (phone REGEXP '^[0-9+ -]{8,15}$')
```

```
);
```

```
ALTER TABLE PetOwner ADD UNIQUE (user_id);
```

Each owner in this table has a unique *owner\_id*, and *user\_id* links them to a user account with a foreign key using the constraint 'ON DELETE RESTRICT' to maintain referential integrity. Next, *name* is NOT NULL and stores the pet owner's name. Moreover, *phone* has a CHECK constraint which validates the phone number format; the check constraint ensures that any value inserted into a column meets a certain condition. In this case, the criteria for the phone attribute are to match the pattern which only has characters (digits 0-9, plus (+), or minus (-)), and the string must be 8-15 characters long. Lastly, *address* is optional, and the *user\_id* has a UNIQUE constraint which prevents one user from having multiple pet owner records.

### C. Veterinarian table

```
CREATE TABLE Veterinarian (  
  vet_id INT AUTO_INCREMENT PRIMARY KEY,  
  user_id INT NOT NULL,  
  name VARCHAR(100) NOT NULL,  
  specialization VARCHAR(100),  
  phone VARCHAR(20),  
  schedule VARCHAR(300),  
  FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE RESTRICT,  
  CHECK (phone REGEXP '^[0-9+-]{8,15}$')  
);
```

```
ALTER TABLE Veterinarian ADD UNIQUE (user_id);
```

Within the veterinarian table, each vet has a unique *vet\_id*, and the *user\_id* links to a user account, like in the petOwner table to prevent orphaned records. *Name* is NOT NULL, *specialisation* and *schedule* are optional fields. Like in the petOwner table, *phone* has a CHECK constraint to validate the format, and the *user\_id* has a UNIQUE constraint to ensure a user cannot have multiple veterinarian accounts.

### D. Clinic table

```
CREATE TABLE Clinic (  
  clinic_id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(100) NOT NULL,  
  address VARCHAR(300) NOT NULL,  
  phone VARCHAR(20),
```

```
CHECK (phone REGEXP '^[0-9+-]{8,15}$')
);
```

The *clinic\_id* uniquely identifies each clinic. *Name* and *address*' constraints are NOT NULL to make sure that important information about the clinic is stored, and *phone* like the two tables above has a CHECK constraint to enforce valid formatting.

#### E. Pet table

```
CREATE TABLE Pet (
    pet_id INT AUTO_INCREMENT PRIMARY KEY,
    owner_id INT,
    name VARCHAR(100) NOT NULL,
    species VARCHAR(50),
    breed VARCHAR(50),
    age INT,
    gender ENUM('Male', 'Female', 'Unknown') DEFAULT 'Unknown'
);
```

The pet table stores data for each pet registered in the system. The primary key *pet\_id* uniquely identifies every pet. The *owner\_id* column links the pet to its owner, while name, species, breed, age, and gender describe the pet's basic characteristics. The name attribute is NOT NULL to ensure that every pet has a recorded name and gender uses an ENUM type to keep the value consistent.

#### F. Appointment table

```
CREATE TABLE Appointment (
    appointment_id INT AUTO_INCREMENT PRIMARY KEY,
    pet_id INT,
    vet_id INT,
    clinic_id INT,
    appointment_date DATETIME NOT NULL,
    status ENUM('Scheduled', 'Completed', 'Cancelled', 'Follow-up
Needed') DEFAULT 'Scheduled'
);
```

The appointment table records scheduled visits between pets and veterinarians at a specific clinic. The primary key *appointment\_id* uniquely identifies each appointment. Columns *pet\_id*, *vet\_id*, and *clinic\_id* reference the pet, veterinarian and clinic involved in the appointment. The *appointment\_date* attribute is NOT NULL to guarantee a

defined date and time, while status uses an ENUM to limit the appointment state to valid options.

#### G. TreatmentRecord table

```
CREATE TABLE TreatmentRecord (
    record_id INT AUTO_INCREMENT PRIMARY KEY,
    diagnosis TEXT,
    treatment TEXT,
    prescription TEXT,
    treatment_date DATE NOT NULL DEFAULT (CURRENT_DATE)
);
```

The TreatmentRecord table stores information about medical treatments given to pets. The primary key record\_id uniquely identifies each treatment record. The columns of diagnosis, treatment, and prescription describe the medical actions taken, while treatment\_date records when the treatment occurred and is set to NOT NULL with a default of the current data.

Entity	Unnormalised Form	1NF	2NF	3NF
User	User(user_id, username, password, role, email, created_at)	Already atomic (no repeating values)	user_id - all attributes depend on this PK	All non-key attributes depend only on user_id
petOwner	PetOwner(owner_id, name, phone, address)	All attributes are atomic	owner_id - all attributes depend on this PK	Phone, address, and name depend only on owner_id
Veterinarian	Veterinarian(vet_id, user_id, name, specialisation, phone, schedule)	All attributes are atomic	vet_id - all attributes depend on this PK	All non-key attributes depend only on vet_id
Clinic	Clinic(clinic_id, name, address, phone)	All attributes are atomic	clinic_id - all attributes depend on this PK	All non-key attributes depend only on clinic_id
Pet	Pet(pet_id, owner_id, name,	All attributes are atomic	pet_id - all attributes	All non-key attributes

	species, breed, age, gender)		depend on this PK	depend only on pet_id
Appointment	Appointment(appoin tment_id, pet_id, vet_id, clinic_id, appointment_date, status)	All attributes are atomic	appointmen t_id - all attributes depend on this PK	All non-key attributes depend only on appointment_id
TreatmentRec ord	TreatmentRecord(re cord_id, diagnosis, treatment, prescription, treatment_date)	All attributes are atomic	record_id - all attributes depend on this PK	All non-key attributes depend only on record_id