

Veterinary Management System

Team 10

Members: Rithvika, Vaishnavi, Harshika, Vishakha

1.) Mini-World Description

The **Veterinary Management System** models the daily operations and relationships within a busy veterinary clinic, where a range of services are provided to keep pets healthy and happy. This world includes animals of various species, their dedicated owners, a team of skilled veterinarians, and supportive clinic staff like nurses and lab technicians. The database tracks every aspect of care, from scheduling appointments and administering treatments to managing inventory and billing, creating a unified system that ensures each pet receives personalized, efficient attention.

2.) Purpose of the Database

The database is designed to streamline the clinic's operations, offering a centralized, organized platform to manage appointments, track treatments, monitor animal health histories, and maintain medicine supplies. By storing and linking information across entities, the database not only improves accuracy in record-keeping but also enhances clinic workflow, ensuring that all essential data is readily accessible. With its detailed tracking of treatments, medications, and payments, the system helps the clinic provide the highest quality of care and personalized service to its patients.

3.) Users of the Database

- **Veterinarians:** Use the database to view animal health records, manage treatment plans, prescribe medications, and log detailed information on each visit and follow-up.
- **Clinic Staff:** Support the clinic's operations by handling appointments, managing inventory, assisting in treatments, and ensuring smooth communication with pet owners.
- **Clinic Administrators:** Utilize the database for high-level oversight, including billing, inventory control, and managing supplier relationships, to maintain an efficient workflow.
- **Pet Owners:** Benefit indirectly by experiencing seamless service, receiving timely appointment reminders, and having access to their pet's comprehensive health history and treatment records.

This system empowers all users to work collaboratively, keeping each pet's welfare at the heart of every interaction, and offering owners the peace of mind that their pets are in well-organized, capable hands.

4.) What the users will do with the database

The database will empower each user group with essential tools and access to manage and streamline veterinary clinic operations effectively. **Veterinarians** will use the database to access comprehensive animal health records, update treatment plans, prescribe medications, and log details from each visit or follow-up, ensuring continuity in animal care. **Clinic staff** will rely on the system to coordinate appointments, manage inventory, assist veterinarians during treatments, and facilitate efficient communication with pet owners. **Clinic administrators** will leverage the database for a broader view of clinic operations, handling billing, overseeing inventory levels, and managing supplier relationships to keep supplies consistent and costs controlled. Although **pet owners** do not directly interact with the database, they benefit from its structured records by receiving well-organized services, timely appointment reminders, and reliable access to their pet’s treatment history. Collectively, these interactions make the database an invaluable tool for ensuring quality animal care, efficient clinic management, and a seamless experience for all users.

5.) Database Requirements

Entity Types:

Strong Entities:

1)Animal:

| Attribute | Data Type | Domain (if applicable) | Description |
|-----------|---------------|------------------------------|---|
| Animal_ID | INT | Positive integers | Primary key; unique identifier for each animal in the system. |
| Name | VARCHAR(50) | Text | Name of the animal, typically given by the owner. |
| Species | VARCHAR(20) | {‘dog’, ‘cat’, ‘bird’, etc.} | Species type of the animal (e.g., dog, cat, bird). |
| Breed | VARCHAR(50) | Text | Specific breed of the animal (e.g., Golden Retriever for dogs). |
| Gender | CHAR(1) | {‘M’, ‘F’} | Gender of the animal; M for Male, F for Female. |
| Age | INT | Positive integers | Age of the animal in years. |
| Weight | DECIMAL (5,2) | Positive decimal values | Current weight of the animal, measured in kilograms. |

| | | | |
|-----------|-------------------------------------|--|---|
| Height | DECIMAL(5,2) | Positive decimal values (e.g., 0.01 to 999.99) | The height of the animal, typically measured in centimeters or inches, depending on clinic standards. |
| Allergies | VARCHAR(1000) or TEXT (Multivalued) | Specific allergens such as foods or agents | Stores a list of known allergies for each animal, which may include specific medications, foods, or environmental factors (e.g., pollen, dust). |
| Owner_ID | INT | Positive integers | Foreign key referencing Owner_ID in the Owner table, associating each animal with its owner. |

Derived Attributes:

Age_in_Months: Derived from the Age attribute by converting years to months (Age * 12).

BMI: Calculated using Weight and an additional attribute Height if available (e.g., Weight / (Height²)).

2)Owner:

| Attribute | Data Type | Domain (if applicable) | Description |
|--------------|-------------------------|--------------------------------|---|
| Owner_ID | INT | Positive integers | Primary key; unique identifier for each pet owner in the system. |
| Name | Composite attribute | Composite: (VARCHAR (50) each) | Full name of the owner, split into three parts: First_Name, Middle_Name, and Last_Name. |
| Address | Composite attribute | Composite: (VARCHAR and INT) | Physical address of the owner, split into H_No, Stree_No, City, Pin_Code |
| Phone_Number | INT (Can never be NULL) | Valid phone number format | The primary phone number for the owner, formatted for international codes if needed. |

| | | | |
|-------|--------------|--------------------|--------------------------------------|
| Email | VARCHAR(100) | Valid email format | Primary email address for the owner. |
|-------|--------------|--------------------|--------------------------------------|

Other attributes that could serve as unique identifiers (candidate keys) are:

- **Email:** If each owner is required to register with a unique email address, Email could act as a unique alternate key.
- **Phone_Number:** If each owner has a unique phone number (and phone numbers are not reused), Phone_Number could be a unique attribute.
- **Combination of Name and Address:** In some cases, the combination of First_Name, Last_Name, and full Address (e.g., H_No, Street_No, City, Pin_Code) might serve as a unique identifier if it's guaranteed to be unique per owner.

3)Medicine Inventory:

| Attribute | Data Type | Domain (if applicable) | Description |
|-------------------|--------------------------------------|--|---|
| Medicine_ID | INT | Positive integers | Primary key; unique identifier for each medicine. |
| Medicine_Name | VARCHAR(100) | Text | Name of the medicine. |
| Storage_Locations | VARCHAR(50) or TEXT (Multivalued) | Location types (e.g., 'main storage', 'secondary storage', 'refrigerated storage') | Stores a list of different storage locations where a specific medicine is kept, depending on inventory management needs and temperature requirements. |
| Type | VARCHAR(50) | {'vaccine', 'antibiotic', etc} | Type of medicine, limited to categories like vaccine, antibiotic, etc. |
| Expiration_Date | DATE | Valid Dates | Expiration date of the medicine. |
| Current_Stock | INT | Positive integers | Part of Stock Levels; indicates the current number of units in stock. |
| Reorder_Level | INT | Positive integers | Part of Stock Levels; indicates the minimum stock level at which |

| | | | |
|--|--|--|--------------------------|
| | | | reordering should occur. |
|--|--|--|--------------------------|

Derived Attributes:

Days_to_Expiration: Calculated by subtracting CURRENT_DATE from the Expiration_Date to determine how many days remain until the medicine expires.

Is_Reorder_Needed: Derived based on a comparison between Current_Stock and Reorder_Level, indicating whether a reorder is necessary.

4)Supplier

| Attribute | Data Type | Domain (if applicable) | Description |
|--------------|------------------------|---------------------------------|--|
| Supplier_ID | INT | Positive integers | Primary key; unique identifier for each supplier |
| Name | VARCHAR(100) | Text | Name of the supplier. |
| Address | Composite attribute | Composite: (VARCHAR and INT) | Physical address of the owner, split into H_No, Stree_No, City, Pin_Code |
| Contact_Info | Multi-valued attribute | (VARCHAR and INT) | The owner's contact information such as phone number and email ID. |

5)Veterinarian (with a subclass for Specialist Veterinarians and Emergency Veterinarians):

| Attribute | Data Type | Domain (if applicable) | Description |
|--------------|---------------------|-----------------------------------|---|
| Vet_ID | INT | Positive integers | Primary key; unique identifier for each veterinarian. |
| Name | Composite attribute | Composite: (VARCHAR (50) each) | Full name of the vet, split into three parts: First_Name, Middle_Name, and Last_Name. |
| Contact_Info | VARCHAR(100) | Valid phone number/email format | Contact information (can be a primary phone number or email address). |

| | | | |
|---------------------|-------------|----------------------------------|--|
| Years_of_Experience | INT | Positive integers | Total years of experience the veterinarian has |
| Specialization | VARCHAR(50) | {‘surgery’, ‘dermatology’, etc.} | Area of specialization, e.g., surgery or dermatology; specific to Specialist Veterinarians only. |

a) Subclass: Specialist Veterinarian:

Additional attributes:

- Certification_Type: Type or level of certification for the specialization (e.g., board-certified, ACVIM certified).
- Specialist_Field : Specific field of specialization, such as oncology or orthopaedics.

b) Emergency Veterinarian (Subclass)

Additional Attributes:

Emergency_Certificate: Certification specific to emergency medicine, including issuing organization. (e.g., ACVECC Diplomate

Availability: Days and times the veterinarian is available for emergencies (e.g., Weekends, Night Shift).

Response_Time: Estimated response time for emergencies (e.g., 15 mins, 30 mins).

6)Appointment

| Attribute | Data Type | Domain (if applicable) | Description |
|-----------------|-------------|--|---|
| Appointment_ID | INT | Positive integers | Primary key; unique identifier for each appointment. |
| Date_Time | DATETIME | Valid date and time format | The date and time of the appointment. |
| AppointmentType | VARCHAR(50) | {‘check-up’, ‘surgery’, ‘vaccination’} | Type of appointment, restricted to specific types such as check-up, surgery, or vaccination |
| Status | VARCHAR(20) | {‘scheduled’, ‘completed’, ‘cancelled’, ‘rescheduled’} | Current status of the appointment. |

| | | | |
|-----------|-----|-------------------|--|
| Vet_ID | INT | Positive integers | Foreign key referencing the Vet_ID in the Veterinarian table, identifying the veterinarian responsible. |
| Animal_ID | INT | Positive integers | Foreign key referencing the Animal_ID in the Animal table, linking the appointment to a specific animal. |

7)Clinic Staff

| Attribute | Data Type | Domain (if applicable) | Description |
|----------------|---------------------|--------------------------------------|--|
| Staff_ID | INT | Positive integers | Primary key; unique identifier for each clinic staff member. |
| Name | Composite attribute | Composite: (VARCHAR (50) each) | Full name of the vet, split into three parts: First_Name, Middle_Name, and Last_Name. |
| Type | VARCHAR(20) | {‘Lab Technician’, ‘Nurse’} | Role or type of clinic staff, restricted to values like Lab Technician or Nurse. |
| Contact_Info | VARCHAR(100) | Valid phone number or email format | Primary contact information for the staff member (e.g., phone number or email). |
| Shift | VARCHAR(20) | {‘Morning’, ‘Evening’, ‘Night’} | Shift assignment for the staff member (e.g., Morning, Evening, Night). |
| Specialization | VARCHAR(50) | Text | Area of specialization, if applicable, such as Surgical Nurse or Pathology Technician. |

Weak Entities

1) Treatments:

| Attribute | Data Type | Domain (if applicable) | Description |
|------------------|--------------|-------------------------------------|---|
| Treatment_Type | VARCHAR(50) | {‘surgery’, ‘vaccination’, etc.} | Partial key; type of treatment, such as surgery, vaccination, etc. |
| Medication_ID | INT | Valid date and time format | Foreign key referencing Medication table, identifying the medication prescribed. |
| Dosage | VARCHAR(50) | Text | Dosage prescribed for the treatment, e.g., 50mg. |
| Duration | VARCHAR(50) | Text | Duration for which the medication needs to be taken, e.g., 7 days. |
| Side_Effects | VARCHAR(100) | Text | Known side effects of the prescribed medication. |
| Animal_ID | INT | Positive integers | Foreign key referencing the Animal_ID in the Animal table. |
| VetID | INT | Positive integers | Foreign key referencing Veterinarian table; veterinarian administering the treatment. |
| OwnerID | INT | Positive Integers | Foreign key referencing Owner table; owner owning a particular animal. |
| FollowUpNeeded | BOOLEAN | {TRUE, FALSE} | Indicates if a follow-up appointment is required. |
| DateAdministered | DATE | Valid dates | Date the treatment was given. |

2)Billing

A Payment Record depends on an Appointment or a Service rendered to an animal, along

with a Pet Owner. The payment is tied to a specific transaction related to veterinary services, so it cannot exist independently.

| Attribute | Data Type | Domain (if applicable) | Description |
|-----------------|---------------|-------------------------------|---|
| Date_of_Payment | DATE | Valid dates | Partial key; date on which the payment was made. |
| Amount_Paid | DECIMAL(10,2) | Positive decimal values | The amount of money paid for the service or treatment. |
| Payment_Method | VARCHAR(50) | {‘cash’, ‘credit card’, etc.} | Payment method used by the owner, e.g., cash, credit card, debit. |
| Appointment_ID | INT | Positive integers | Foreign key referencing the Appointment table, identifying the specific appointment billed. |
| Treatment_ID | INT | Positive integers | Foreign key referencing the Treatment table, identifying a specific treatment if applicable.. |
| Owner_ID | INT | Positive integers | Foreign key referencing the Owner table, linking the billing record to the pet owner. |

Derived Attributes:

Total_Amount_Due: Calculated as the sum of individual Amount_Paid entries associated with a specific Owner_ID over a period.

Overdue_Days: Derived by subtracting the Date_of_Payment from the due date, showing how many days a payment is overdue.

6.) Relationship Types

Degree: Binary (2)

1). Owns Relationship:

- Participating Entity Types: Owner, Animal

- Cardinality Ratio: One-to-Many (1,N)
- Participation Constraints:
- Owner: Total (Every owner must have at least 1 pet)
- Animal: Total (Every animal must have an owner)
- (Min, Max) Constraints:
- Owner: (1, N) - An owner might own 1 to many pets.
- Animal: (1, 1) - Each animal must be owned by exactly one owner.

2). Schedules Relationship:

- Participating Entity Types: Veterinarian, Appointment
- Cardinality Ratio: One-to-Many (1,N)
- Participation Constraints:
- Veterinarian: Partial (Not every veterinarian may always have appointments)
- Appointment: Total (Every appointment must be scheduled with a veterinarian)
- (Min, Max) Constraints:
- Veterinarian: (0, N) - A veterinarian may have no appointments up to many
- appointments.
- Appointment: (1, 1) - Each appointment must be scheduled with one veterinarian.

3). SuppliesMedicine:

- Participating Entity Types: Supplier, Medicine Inventory
- Cardinality Ratio: One-to-Many (1, N)
- Participation Constraints:
- Supplier: Partial (Not all suppliers may provide all medicines in the inventory).
- Medicine Inventory: Total (Every medicine in the inventory must have at least one supplier).
- (Min, Max) Constraints:
- Supplier: (0, N) – A supplier can supply zero to many types of medicine.
- Medicine Inventory: (1, 1) – Each medicine in the inventory must have exactly one supplier.

Attributes of SuppliesMedicine:

- Quantity: The amount of each specific medicine provided by the supplier.

4). Mentorship (Recursive)

- Description: Experienced veterinarians can mentor newer veterinarians within the clinic.
- Participating Entity Type: Veterinarian (in the roles of mentor and mentee)
- Cardinality Ratio: One-to-Many (One veterinarian can mentor multiple mentees, but each mentee has only one mentor).

- Participation Constraints:
- Mentor Veterinarian: Partial (not all veterinarians are mentors).
- Mentee Veterinarian: Total (All veterinarians have mentors).
- (Min, Max) Constraints:
- Mentor Veterinarian: (0, N) - A mentor can have none or many mentees.
- Mentee Veterinarian: (1, 1) - Each mentee must have one mentor.

Degree: Ternary (3)

1.) Payment_Record

Participating Entities: Owner, Billing, Treatment

Cardinality Ratio: Many-to-Many-to-Many (M:N:P)

- Owner can have multiple billing records associated with different treatments.
- Billing entries can involve different treatments for various owners.
- Treatment can have payments from different owners depending on the billing structure.

Participation Constraints:

- Owner: Total (Every payment record must be linked to an owner who made the payment).
- Billing: Total (Every billing entry should involve in a specific treatment, as some may cover general services).
- Treatment: Total (Treatments have a payment record if billed).
- (Min, Max) Constraints:
- Owner: (1, N) – Each owner can be associated with one or more billing entries covering various treatments.
- Billing: (1, N) – Each billing entry can be associated with more specific treatments, whether it be checkup, medication, or surgery
- Treatment: (1, N) – Each treatment must have been paid for a treatment.

This relationship indicates which treatments have been paid for by each owner, capturing the connection between an Owner, a Billing entry, and a specific Treatment provided to an animal.

2.) AssistsInAppointment

- Description: This relationship indicates which clinic staff assist veterinarians in specific appointments. The clinic staff may take on different roles during the appointment, such as assisting in surgeries, administering medication, or preparing the animal for treatment.

- Participating Entity Types: Clinic Staff (provides support during the appointment), Veterinarian (conducts the appointment and oversees procedures), Appointment (the scheduled appointment for a specific animal)
- Cardinality Ratio: Many-to-Many-to-Many (M: N:P)
 - A single clinic staff member can assist in multiple appointments and support multiple veterinarians.
 - A veterinarian may have various clinic staff members assisting in different appointments.
 - An appointment may involve multiple clinic staff members supporting one veterinarian.
- Participation Constraints:
 - Clinic Staff: Partial (not all appointments require support from clinic staff; some may be managed solely by the veterinarian).
 - Veterinarian: Partial (A veterinarian may or may not oversee an appointment).
 - Appointment: Total (each appointment requires a veterinarian and, in some cases, clinic staff support).
- (Min, Max) Constraints:
 - Clinic Staff: (0, N) – Clinic staff members may assist in zero or many appointments.
 - Veterinarian: (0, N) – Each veterinarian may oversee one or multiple appointments.
 - Appointment: (1, N) – Each appointment requires at least one veterinarian and may have multiple clinic staff members.

This AssistsInAppointment relationship helps track which clinic staff members were involved in supporting specific appointments alongside veterinarians, ensuring accurate records for each visit and supporting personnel management.

Degree: Quaternary(4)

1.) AdministersTreatment

Participating Entities: Veterinarian, Animal., Clinic Staff, Treatment

Cardinality Ratio: Many-to-Many-to-Many-to-Many (M:N:P:Q)

- Veterinarian: Each veterinarian can administer multiple treatments to multiple animals, often with the assistance of various clinic staff members.
- Animal: Each animal can receive multiple treatments from different veterinarians and clinic staff.

- Clinic Staff: Each clinic staff member can assist with multiple treatments involving different veterinarians and animals.
- Treatment: Each treatment can involve multiple veterinarians, clinic staff, and animals, depending on the clinic's operations and treatment requirements.

Participation Constraints:

- Veterinarian: Partial (Not every veterinarian may administer every treatment, depending on their specialization).
- Animal: Total (Every administered treatment must involve an animal).
- Clinic Staff: Partial (Not all treatments require the assistance of clinic staff).
- Treatment: Total (Each treatment record must involve a veterinarian, animal, and optionally clinic staff).

(Min, Max) Constraints:

- Veterinarian: (0, N) – A veterinarian may administer zero or more treatments to animals.
- Animal: (1, N) – Each animal can receive one or more treatments.
- Clinic Staff: (0, N) – A clinic staff member may assist with zero or more treatments.
- Treatment: (1, N) – Each treatment administered requires one record with a specific animal and veterinarian.

This relationship records the treatments administered to animals, detailing the roles of both the veterinarian and clinic staff involved in the process. It ensures each treatment record captures all personnel involved and the animal receiving the treatment.

7.) Functional Requirements:

1. Retrieval:

(a) Query Functions:

Selection:

- Retrieve complete data for appointments scheduled for a specific veterinarian.
Example Query: `SELECT * FROM Appointment WHERE Vet_ID = <specific_vet_id>;`
- Retrieve all animals belonging to a specific owner.
Example Query: `SELECT * FROM Animal WHERE Owner_ID = <specific_owner_id>;`

Projection Query:

- List only the Name and Breed of all animals that are of the species "dog."
Example Query:
`SELECT Name, Breed`

```
FROM Animal
WHERE Species = 'dog';
```

- Retrieve the names of all medicines that are nearing their reorder level.

Example Query:

```
SELECT Medicine_Name FROM Medicine WHERE Current_Stock <= Reorder_Level;
```

Aggregate Function:

- Retrieve the maximum years of experience among all veterinarians.

Example Query:

```
SELECT MAX(Years_of_Experience) AS Max_Experience
FROM Veterinarian;
```

- Find the minimum stock level for each type of medicine (e.g., vaccine, antibiotic).

Example Query:

```
SELECT Type, MIN(Current_Stock) AS Min_Stock_Level
FROM Medicine
GROUP BY Type;
```

Search Query (Partial Text Match):

- Find all animals whose names contain the substring "Max" (e.g., "Max", "Maxwell").

Example Query:

```
SELECT *
FROM Animal
WHERE Name LIKE '%Max%';
```

- List all veterinarians specializing in any field that includes the term "surgery" (e.g., "orthopedic surgery").

```
SELECT First_Name, Last_Name, Specialization
FROM Veterinarian
WHERE Specialization LIKE '%surgery%';
```

(b) Analysis Queries:

1. Report on Total Amount Billed per Owner for All Treatments and Appointments

This report calculates the total amount billed to each owner by combining Billing, Owner, and Appointment tables. It includes the total payments made for both treatments and appointments.

Example Query:

```
SELECT O.Owner_ID, O.First_Name, O.Last_Name, SUM(B.Amount_Paid) AS Total_Billed
FROM Owner O
JOIN Billing B ON O.Owner_ID = B.Owner_ID
JOIN Appointment A ON B.Appointment_ID = A.Appointment_ID
GROUP BY O.Owner_ID, O.First_Name, O.Last_Name;
```

Purpose: Provides an overview of total payments made by each owner, covering both treatments and general appointments.

2. Report on Average Treatment Cost by Treatment Type

This query calculates the average cost of each type of treatment by joining Treatments and Billing entities.

```
SELECT T.Treatment_Type, AVG(B.Amount_Paid) AS Average_Cost
FROM Treatments T
JOIN Billing B ON T.Treatment_ID = B.Treatment_ID
GROUP BY T.Treatment_Type;
```

Purpose: Offers insights into the average cost per treatment type (e.g., surgery, vaccination), helping identify cost trends for different services.

3. Report on Average Weight of Animals by Species Treated by Each Veterinarian

This report calculates the average weight of animals treated by each veterinarian, grouped by species. It joins the Animal, Treatment, and Veterinarian tables.

```
SELECT V.Vet_ID, V.First_Name, V.Last_Name, A.Species, AVG(A.Weight) AS
Avg_Animal_Weight
FROM Veterinarian V
JOIN Treatments T ON V.Vet_ID = T.VetID
JOIN Animal A ON T.Animal_ID = A.Animal_ID
GROUP BY V.Vet_ID, V.First_Name, V.Last_Name, A.Species;
```

Purpose: Provides information on the types and sizes of animals each veterinarian typically treats, which could help identify treatment specialties or requirements for different species.

2. Modification:

(a) Insertion:

1.) Add a new appointment for an animal, ensuring the Animal_ID and Vet_ID are valid and exist in the Animal and Veterinarian tables, respectively.

Query:

```
INSERT INTO Appointment (Appointment_ID, Date_Time, AppointmentType, Status, Vet_ID, Animal_ID)
```

```
VALUES (101, '2023-11-15 10:30:00', 'check-up', 'scheduled', 2002, 3001);
```

Integrity Check: This will check that:

- Vet_ID (2002) exists in the Veterinarian table.
- Animal_ID (3001) exists in the Animal table.
- If either does not exist, the insertion will fail due to foreign key constraint violations.

2.) Insert a New Animal Record Insert a new animal, ensuring that the Owner_ID exists in the Owner table.

Query:

```
INSERT INTO Animal (Animal_ID, Name, Species, Breed, Gender, Age, Weight, Owner_ID)
```

```
VALUES (5003, 'Bella', 'dog', 'Labrador', 'F', 3, 24.5, 1001);
```

Integrity Check: Confirms that Owner_ID (1001) exists in the Owner table. If it does not, the insertion will fail. Animal ID should be unique. Animal ID should be unique.

b) Update Operation:

1.) Update the status of an appointment once it has been completed. Ensure that the Appointment_ID exists before updating.

Query:

```
UPDATE Appointment
```

```
SET Status = 'completed'
```

```
WHERE Appointment_ID = 101;
```

Integrity Check: This will ensure that Appointment_ID (101) exists in the Appointment table. If it does not, no rows will be updated.

2.) Increase Veterinarian's Years of Experience

Increment the years of experience for a veterinarian after a performance review, ensuring the veterinarian exists in the system.

Query:

```
UPDATE Veterinarian
```

```
SET Years_of_Experience = Years_of_Experience + 1
```

```
WHERE Vet_ID = 2002;
```

Integrity Check: Confirms Vet_ID (2002) exists in Veterinarian. This operation increases Years_of_Experience by 1, ensuring valid updates for existing veterinarians only.

(c) Delete Operation:

1.) Delete Medicine Inventory Entry for Expired Medicine

Remove a medicine from inventory if it has expired, ensuring the Expiration_Date is before today.

Query:

```
DELETE FROM Medicine
```

```
WHERE Expiration_Date < CURRENT_DATE;
```

Integrity Check: Only deletes medicines whose Expiration_Date is in the past, ensuring valid medicine records are retained.

2.) Delete Appointments Scheduled in the Past for Canceled Status

Remove records of appointments that were canceled and are in the past, as they no longer require tracking.

Query:

```
DELETE FROM Appointment
```

```
WHERE Status = 'canceled' AND Date_Time < CURRENT_DATE;
```

Integrity Check: Only deletes past appointments that were canceled, keeping the record of other past and future appointments intact.