```
-- Create Animal table
CREATE TABLE Animal (
    Animal_ID INTEGER PRIMARY KEY,
    Name VARCHAR2(50) NOT NULL,
    Arrival_Date DATE NOT NULL,
    Behavior_Notes CLOB,
    Treatment_ID INTEGER,
    Med_ID INTEGER
);

-- Create Volunteer table
CREATE TABLE Volunteer (
    Volunteer_ID INTEGER PRIMARY KEY,
    Name VARCHAR2(50) NOT NULL,
    Role VARCHAR2(50),
    Availability CLOB
);

-- Create Employee table
CREATE TABLE Employee (
    Employee_ID INTEGER PRIMARY KEY,
    Name VARCHAR2(50) NOT NULL,
    Role VARCHAR2(50),
    Certifications CLOB,
    Availability CLOB
);

-- Create Emergency Protocol table
CREATE TABLE Emergency_Protocol (
    Protocol_ID INTEGER PRIMARY KEY,
    Description CLOB NOT NULL
);

-- Create Medication table
CREATE TABLE Medication (
    Med_ID INTEGER PRIMARY KEY,
    Name VARCHAR2(50) NOT NULL,
    Type VARCHAR2(50),
    Usage CLOB
);

-- Create Caretaker table (references Animal)
CREATE TABLE Caretaker (
    Caretaker_ID INTEGER PRIMARY KEY,
    Animal_ID INTEGER NOT NULL,
    FOREIGN KEY (Animal_ID) REFERENCES Animal(Animal_ID)
);

-- Create Enclosure table (references Animal)
CREATE TABLE ENCLOSURE (
    enclosure_id NUMBER PRIMARY KEY,
    capacity NUMBER NOT NULL,
    caretaker_id NUMBER,
    FOREIGN KEY (caretaker_id) REFERENCES CARETAKER(caretaker_id)
);

-- Create Species table (references Emergency_Protocol)
CREATE TABLE Species (
    Species_ID INTEGER PRIMARY KEY,
```

```
    Name VARCHAR2(50) NOT NULL,
    Diet_Plan CLOB,
    Enclosure_Requirements CLOB,
    Breeding_Program_Status CHAR(1) CHECK (Breeding_Program_Status IN ('Y', 'N')),
    Protocol_ID INTEGER,
    FOREIGN KEY (Protocol_ID) REFERENCES Emergency_Protocol(Protocol_ID)
);

-- Create Conservation table (references Species)
CREATE TABLE Conservation (
    Program_ID INTEGER PRIMARY KEY,
    Name VARCHAR2(50) NOT NULL,
    Partner_Organization VARCHAR2(100),
    Species_ID INTEGER,
    FOREIGN KEY (Species_ID) REFERENCES Species(Species_ID)
);

-- Create Visitor table
CREATE TABLE Visitor (
    Visitor_ID INTEGER PRIMARY KEY,
    Name VARCHAR2(50) NOT NULL,
    Membership VARCHAR2(20)
);

-- Create Financial Transaction table (references Visitor)
CREATE TABLE Financial_Transaction (
    Transaction_ID INTEGER PRIMARY KEY,
    Type VARCHAR2(20) NOT NULL,
    Amount DECIMAL(10, 2) CHECK (Amount >= 0),
    Transaction_Date DATE NOT NULL,  -- Renamed to avoid 'Date' conflict
    Visitor_ID INTEGER NOT NULL,
    FOREIGN KEY (Visitor_ID) REFERENCES Visitor(Visitor_ID)
);

-- Create Donation table (references Financial_Transaction)
CREATE TABLE Donation (
    Donation_ID INTEGER PRIMARY KEY,
    Donor VARCHAR2(50) NOT NULL,
    Amount DECIMAL(10, 2) CHECK (Amount >= 0),
    Donation_Date DATE NOT NULL,
    Transaction_ID INTEGER NOT NULL,
    FOREIGN KEY (Transaction_ID) REFERENCES Financial_Transaction(Transaction_ID)
);


-- Create Veterinarian table (references Employee and Medication)
CREATE TABLE Veterinarian (
    Vet_ID INTEGER PRIMARY KEY,
    Medical_License VARCHAR2(20),
    Animal_ID INTEGER,
    Med_ID INTEGER,
    FOREIGN KEY (Animal_ID) REFERENCES Animal(ANIMAL_ID),
    FOREIGN KEY (Med_ID) REFERENCES Medication(Med_ID)
);


-- Create Treatment table
CREATE TABLE Treatment (
    Treatment_ID INTEGER PRIMARY KEY,
```

```
    Treatment_Name VARCHAR2(50),
    Treatment_Date DATE NOT NULL,
    Animal_ID INTEGER,
    FOREIGN KEY (Animal_ID) REFERENCES Animal(Animal_ID)
)
PARTITION BY RANGE (Treatment_Date) (
    PARTITION treatment_2023 VALUES LESS THAN (TO_DATE('2024-01-01', 'YYYY-MM-
DD')),
    PARTITION treatment_2024 VALUES LESS THAN (TO_DATE('2025-01-01', 'YYYY-MM-
DD')),
    PARTITION treatment_2025 VALUES LESS THAN (TO_DATE('2026-01-01', 'YYYY-MM-DD'))
);


-- Create Animal Monitoring table (references Animal and Volunteer)
CREATE TABLE Animal_Monitoring (
    Monitor_ID INTEGER PRIMARY KEY,
    Animal_ID INTEGER NOT NULL,
    Volunteer_ID INTEGER NOT NULL,
    FOREIGN KEY (Animal_ID) REFERENCES Animal(Animal_ID),
    FOREIGN KEY (Volunteer_ID) REFERENCES Volunteer(Volunteer_ID)
);

-- Create Supervision table (references Employee and Volunteer)
CREATE TABLE Supervision (
    Supervision_ID INTEGER PRIMARY KEY,
    Employee_ID INTEGER NOT NULL,
    Volunteer_ID INTEGER NOT NULL,
    FOREIGN KEY (Employee_ID) REFERENCES Employee(Employee_ID),
    FOREIGN KEY (Volunteer_ID) REFERENCES Volunteer(Volunteer_ID)
);

-- Create Animal Responsibility table (references Veterinarian and Animal)
CREATE TABLE Animal_Responsibility (
    Responsibility_ID INTEGER PRIMARY KEY,
    Vet_ID INTEGER NOT NULL,
    Animal_ID INTEGER NOT NULL,
    Assigned_Date DATE NOT NULL,
    FOREIGN KEY (Vet_ID) REFERENCES Veterinarian(Vet_ID),
    FOREIGN KEY (Animal_ID) REFERENCES Animal(Animal_ID)
);


-- Indexes for foreign key columns
CREATE INDEX idx_animal_enclosure ON Enclosure(Animal_ID);

Enclosure will get an error, if so use this than put in the index again
ALTER TABLE Enclosure
ADD CONSTRAINT fk_animal_enclosure
FOREIGN KEY (Animal_ID) REFERENCES Animal(Animal_ID);



CREATE INDEX idx_species_protocol ON Species(Protocol_ID);
CREATE INDEX idx_transaction_visitor ON Financial_Transaction(Visitor_ID);
CREATE INDEX idx_donation_transaction ON Donation(Transaction_ID);
CREATE INDEX idx_animal_monitoring ON Animal_Monitoring(Animal_ID);

-- Indexes for frequently queried columns
```

```
CREATE INDEX idx_animal_name ON Animal(Name);
CREATE INDEX idx_volunteer_name ON Volunteer(Name);
CREATE INDEX idx_employee_name ON Employee(Name);
```