Davis Flowers Project Part 2

A) From Logical Model:

Clinic (clinicNo, name, address, phone, managerNo)
Primary Key clinicNo
Alternate Key phone
Alternate Key managerNo
Foreign Key managerNo **references** Staff(staffNo)

Staff (staffNo, staff_name, staff_address, staff_phone, DOB, salary, position, clinicNo)
Primary Key staffNo
Foreign Key clinicNo **references** Clinic(clinicNo)

Owner (ownerNo, owner_name, owner_address, owner_phone)
Primary Key ownerNo

Pet (petNo, pet_name, pet_DOB, species, breed, color, ownerNo, clinicNo)
Primary Key petNo
Foreign Key ownerNo **references** Owner(ownerNo)
Foreign Key clinicNo **references** Clinic(clinicNo)

Examination (examNo, chief_complaint, description, exam_date, actions_taken, staffNo, petNo)
Primary Key examNo
Foreign Key staffNo **references** Staff(staffNo)
Foreign Key petNo **references** Pet(petNo)

B) Validate using Normalization to 3NF:

dependencies:

clinicNo → name, address, phone, managerNo
staffNo → staff_name, staff_address, staff_phone, DOB, salary, position, clinicNo
ownerNO → owner_name, owner_address, owner_phone
petNo → pet_name, pet_DOB, species, breed, color, ownerNo, clinicNo
examNo→ chief_complaint, description, exam_date, actions_taken, staffNo, petNo

It is already in first normal form, as there are no existing tuples, and each tuple entered will only have one entry per attribute. There are no partial dependencies, as each relationship has only one primary key, so it is already in 2NF. There are no transitive dependencies present as each table only has its primary dependency between the primary key and the rest of the table, and all foreign keys point to their respective relations with no duplication. The table is validated using normalization to 3NF.

C) Validate against user transaction:

When creating a clinic, how do you ensure that there is a managerNo that can be assigned to staff that do not currently work at the clinic?

We can allow clinicNo to be null or unset when creating staff members so that we have at least one unassigned staff member to use as manager. This is assuming that when a clinic is created there is a manager within the initial staff, as all clinics must be managed.

When creating an entry in Pet, how do you register it to a new owner?

We make it so that a pet cannot be registered to an unestablished owner. This currently goes against the multiplicity between owner and pet which states that an owner can own one to many. We can adjust this to have owners exist in the system even if they do not own a pet. That way we can enter owners and not worry about the pet requirement on creation.

When creating an entry in Examination, how do we ensure that there are no inconsistencies between the staff and pet location?

We ensure that Pet(clinicNo) is the same as the staff(clinicNo) when assigning creating an examination.

If the clinic that either the staff member or pet are assigned to changes what happens to the examination?

The examination will only require the same location for both the pet and staff on creation, it will not need to update with the changes. It will be static after creation.

If a pet passes away, what do we do with the relations it is connected to?

The relations it references (Owner and Clinic) do not need to update at all, with the removal of the pet from the system. However, the relation it is referenced in (examinations) needs to be un altered with the removal as to preserve records of examinations.

D) Define integrity constraints

i,ii, iii) Primary Key Constraints and referential integrity

Clinic (clinicNo, name, address, phone, managerNo)
Primary Key clinicNo
Alternate Key phone
Alternate Key managerNo
Foreign Key managerNo **references** Staff(staffNo) ON UPDATE CASCADE ON DELETE SET NULL

Staff (staffNo, staff_name, staff_address, staff_phone, DOB, salary, position, clinicNo)
Primary Key staffNo
Foreign Key clinicNo **references** Clinic(clinicNo) ON UPDATE CASCADE ON DELETE SET NULL

Owner (ownerNo, owner_name, owner_address, owner_phone)
Primary Key ownerNo

Pet (petNo, pet_name, pet_DOB, species, breed, color, ownerNo, clinicNo)
Primary Key petNo
Foreign Key ownerNo **references** Owner(ownerNo) ON UPDATE CASCADE ON DELETE CASCADE
Foreign Key clinicNo **references** Clinic(clinicNo) ON UPDATE CASCADE ON DELETE SET NULL

Examination (examNo, chief_complaint, description, exam_date, actions_taken, staffNo, petNo)
Primary Key examNo
Foreign Key staffNo **references** Staff(staffNo) ON UPDATE CASCADE ON DELETE NO ACTION
Foreign Key petNo **references** Pet(petNo) ON UPDATE CASCADE ON DELETE NO ACTION

iv) require Data

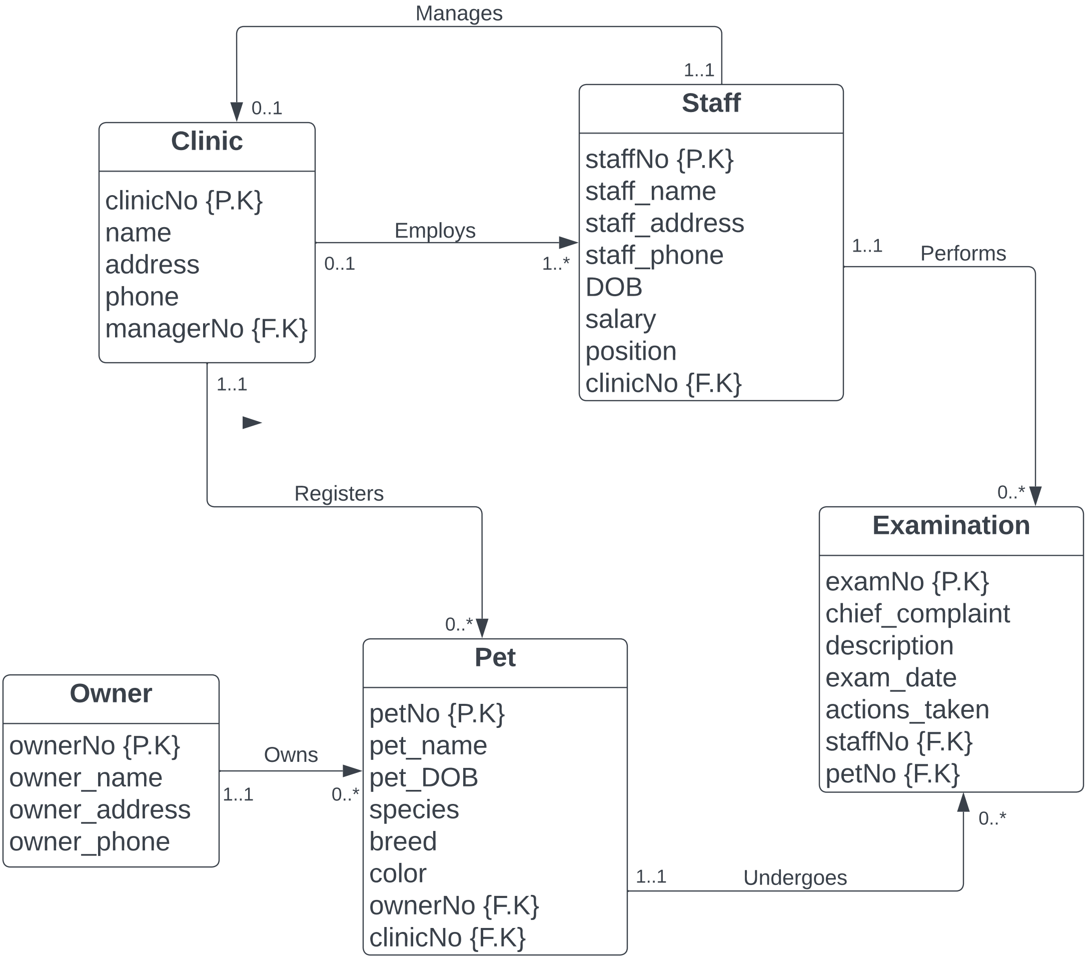
All primary keys are required.
Clinic(name, address, and phone ALL NON NULL)
Staff(staff_name, DOB, salary ALL NON NULL)
Owner(owner_phone NON NULL)
Pet(ownerNo NON NULL)
Examination(date, staffNo, petNo) NON NULL

v) attribute domain constraints

- clinicNo, staffNo, petNo, ownerNo, examNo -- **varchar(8)**
- name, address, staff_name, staff_address, position, owner_name, owner_address, pet_name, species, breed, color, chief_complaint, description, actions_taken – **varchar(255)**
- phone, staff_phone, owner_phone – **varchar(20) CHECK** (phone not like '%[^0-9]%')
- DOB, pet_DOB, exam_date – **DATE**
- Salary – **int(12)**

vi) General constraints - NONE

E) ER Diagram


Manages
1..1
0..1
Clinic
clinicNo {P.K}
name
address
phone
managerNo {F.K}
Staff
staffNo {P.K}
staff_name
staff_address
staff_phone
DOB
salary
position
clinicNo {F.K}
Employs
0..1
1..*
1..1
Performs
1..1
Registers
0..*
0..*
Examination
examNo {P.K}
chief_complaint
description
exam_date
actions_taken
staffNo {F.K}
petNo {F.K}
Pet
petNo {P.K}
pet_name
pet_DOB
species
breed
color
ownerNo {F.K}
clinicNo {F.K}
Owner
ownerNo {P.K}
owner_name
owner_address
owner_phone
Owns
1..1
0..*
0..*
1..1
Undergoes