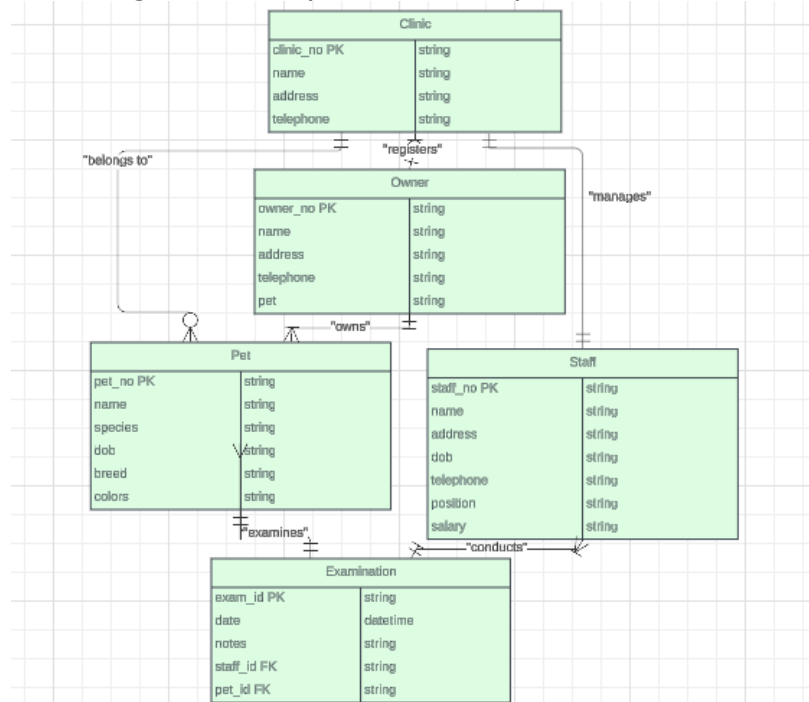

Develop a logical data model based on the following requirements Part 2: (11/20/24) Olga Escoto Balas

a. Derive relations from the conceptual model. The conceptual model I created in part 1 is below for reference and starting point. Based on this, the relations from the conceptual model are as follows:

- **Clinic:** clinicNo (PK), name, address, telephone
- **Staff:** staffNo (PK), name, address, telephone, DOB, position, salary, clinicNo (FK) (will add to the drawing of tables below).
- **Owner:** ownerNo (PK), name, address, telephone, pet.
- **Pet:** petNo (PK), name, DOB, species, breed, color, ownerNo (FK), clinicNo (FK) (will add to the drawing of tables below).
- **Examination:** examNo (PK), dateSeen, actionsTaken, pet_id (FK), staff_id (FK) adding chiefcomplaint, description in Part 2.



b. Validate the logical model using normalization to 3NF.

The relationship I created in part 1 are in 3NF. In my tables, all none-key columns correlate to the primary key indicated.

c. Validate the logical model against 5 user transactions.

(Note: These will be then implemented in 3c).

So since these will be implemented in 3C, from a validation perspective if 5 user transactions were to be implemented, lets see how the tables could handle it:

- Add a new clinic: adding another row in the Clinic table with clinicNo & related fields in that table being populated.
- Register a new pet owner and their pet: using Owner and Pet tables with clinicNo as a foreign key with related information.
- Assign a staff member to manage a clinic: Updates clinicNo in the Staff table. Potentially, update the salary as well and also the position.
- Record an examination for a pet: Requires Examination linked to Pet and Staff. Pet can be linked to Owner and clinic.
- Retrieve all examinations for a specific pet: Run a query where the primary key is used in the Examination table joined with Pet table. This will work.

Based on the walk through, inserting 5 user transitions will not create an issue for any of the tables & the relationships between the tables.

d. Define integrity constraints:

i. Primary key constraints.

- clinicNo for Clinic
- staffNo for Staff
- ownerNo for Owner
- petNo for Pet
- examNo for Examination

ii. Referential integrity/Foreign key constraints.

- clinicNo in Staff = Clinic(clinicNo)
- ownerNo in Pet = Owner(ownerNo)
- clinicNo in Pet = Clinic(clinicNo)
- pet_id in Examination = Pet(petNo)
- staff_id in Examination = Staff(staffNo)

iii. Alternate key constraints (if any).

- None that I can locate.

iv. Required data.

- Primary/Foreign Keys cannot be null. Also, fields such as address such be populated to be able to provide all data elements if asked.

v. Attribute domain constraints.

- Telephone number has to be a number, DOB has to be in a consistent date format.

vi. General constraints (if any).

- One pet per owner, if there are multiple pets per owner they will still receive their own petNo & be linked to the owner that way.
- As indicated in the case study, one staff can manage one clinic.

d. Generate the E-R diagram for the logical level (contains FKs as attributes)

I created this using lucidChart, below is the Part 2 version.

