Doménica Núñez del Arco and Milan Diaz

CSC 423: Database Systems

Design, development, and implementation of a relational database
Part # 2

Develop a logical data model based on the following requirements:

a) Derive relations from the conceptual model

Here is an overview of the relationships obtained in the conceptual data model.

Entity Name	*Relationship*	*Entity Name*	*Relationship Type*
Staff	*Manages*	**Clinic**	1:1
Pet	*IsRegistered*	**Clinic**	1:*
Pet	*Undergoes*	**Examination**	1:*
Owner	*Owns*	**Pet**	1:*
Staff	*Conducts*	**Examination**	1:*
Clinic	*Employs*	**Staff**	1:*

For each 1:* binary relationship, the entity on the "one side" of the relationship is designated as the parent entity and the entity on the "many side" is designated as the child entity. For each 1:1 binary relationship, we identify the parent and child entities for the 1:1 relationship using the participation constraints: the entity with optional participation is the parent entity, and the entity with mandatory participation the child. In both cases, we post a copy of the parent's primary key into the child entity, to act as a foreign key.

Staff (<u>staffNo</u>, staffFName, staffLName, staffAddress, staffDOB, position, salary, clinicNo)

- **Primary Key** staffNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo)

Clinic (<u>clinicNo</u>, clinicName, clinicAddress, clinicTelephone, manager)

- **Primary Key** clinicNo
- **Alternate Key** clinicAddress, **Alternate Key** clinicName, **Alternate Key** clinicTelephone[1]
- **Foreign Key** manager **references** Staff(staffNo)

[1] Following the same assumptions made in the conceptual data model: each clinic has a unique name, address, and telephone.

Pet (petNo, petDOB, species, breed, color, clinicNo, ownerNo)

- **Primary Key** petNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo)
- **Foreign Key** ownerNo **references** Owner(ownerNo)

Owner (ownerNo, ownerFName, ownerLName, ownerAddress, ownerTelephone)

- **Primary Key** ownerNo

Examination (examNo, complaint, description, date, action, petNo, staffNo)

- **Primary Key** examNo
- **Foreign Key** petNo **references** Pet(petNo)
- **Foreign Key** staffNo **references** Staff(staffNo)

b) Validate the logical model using normalization to 3NF

First Normal Form (1NF):
To be in 1NF, a relation needs to have a singular value in the intersection of each row and column. Since that is the case for all of our relations, we say that Staff, Clinic, Pet, Owner, and Examination are all in 1NF.

Second Normal Form (2NF):
A relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on the primary key is in 2NF. In other words, to go from 1NF to 2NF, we must remove partial dependencies. A relation with a single-attribute primary key is automatically in at least 2NF, since there is no possibility of partial dependencies. Therefore, because Staff, Clinic, Pet, Owner, and Examination all have a single primary key, they are all in 2NF.

Third Normal Form (3NF):
A relation is in 3NF if it is in 1NF and 2NF, and no non-primary key attribute is transitively dependent on the primary key. Therefore, to go from 2NF to 3NF, we must remove transitive dependencies, which occur when a non-primary-key attribute depends on another non-primary-key attribute.

1. Staff (staffNo, staffFName, staffLName, staffAddress, staffDOB, position, salary, clinicNo)
A staff member's **salary** depends on their **position** (doctor, nurse, etc.). Consequently, **salary** is transitively dependent on **position**. We will assume pay is consistent across hospitals (a doctor in Hospital A earns the same as a doctor in Hospital B), and therefore, **salary** is not transitively dependent on **clinicNo**. Normalizing, we obtain the following two relations in 3NF:

Staff (staffNo, staffFName, staffLName, staffAddress, staffDOB, position, clinicNo)

- **Primary Key** staffNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo)
- **Foreign Key** position **references** Position(position)

Position (position, salary)

- **Primary Key** position

2. Clinic (clinicNo, clinicName, clinicAddress, clinicTelephone, manager)

A clinic's **clinicName**, **clinicAddress**, **clinicTelephone**, and **manager** are specific to each clinic, so they depend solely on the primary key. There are no transitive dependencies, so the relation stays the same.

3. Pet (petNo, petDOB, species, breed, color, clinicNo, ownerNo)

A pet's **petDOB**, **clinicNo**, and **ownerNo** are specific to each pet, so they depend solely on the primary key. Therefore, there are no transitive dependencies involving these attributes.

A pet's **breed** depends on its **species** (a dog can be a Schnauzer, but a cat cannot), and consequently, **breed** is transitively dependent on **species**. Since each animal has a unique combination of species-breed, these attributes are normalized into their own table as primary keys (**species** alone cannot be a primary key, since a species can have many breeds). An alternative to this approach could be having a specific relation for each animal (Dog relation, Cat relation, etc.), but since we don't know what species will be stored in the database, we will use a general Animal relation.

Additionally, **color** is not transitively dependent on **breed** nor **species**. For example, say pet A is a Labrador dog, while pet B is a Siamese cat - both animals can be brown, despite not being the same breed or species. Moreover, say pet C is also a Labrador dog - it could be black, despite being the same breed and species as pet A. Therefore, **color** is an attribute specific to each pet, and depends solely on **petNo**. Normalizing, we obtain the following two relations in 3NF:

Pet (petNo, petDOB, species, breed, color, clinicNo, ownerNo)

- **Primary Key** petNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo)
- **Foreign Key** ownerNo **references** Owner(ownerNo)
- **Foreign Key** species **references** Animal(species)
- **Foreign Key** breed **references** Animal(breed)

Animal (species, breed)

- **Primary Key** species, breed

Having an Animal relation helps with scalability. Even though the only current attributes are the primary key (species, breed), additional attributes, such as diet, lifeSpan, temperament, etc., could be implemented in the future.

4. Owner (ownerNo, ownerFName, ownerFName, ownerAddress, ownerTelephone)

An owner's **ownerFName**, **ownerFName**, **ownerAddress**, and **ownerTelephone** are specific to each clinic, so they depend solely on the primary key. There are no transitive dependencies, so the relation stays the same.

5. Examination (examNo, complaint, description, date, action, petNo, staffNo)
An examination's **complaint**, **description**, **date**, **petNo** and **staffNo** are specific to each clinic, so they depend solely on the primary key. While it could be argued that **action** is transitively dependent on **complaint** (if a complaint is broken bone, the action is x-ray). However, because each case is unique, and different doctors may take different actions, we will assume **action** is not dependent on **complaint**. There are no transitive dependencies, so the relation stays the same.

To summarize, these are the relations we have after normalization.

Staff (staffNo, staffFName, staffLName, staffAddress, staffDOB, position, clinicNo)

- **Primary Key** staffNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo)
- **Foreign Key** position **references** Position(position)

Position (position, salary)

- **Primary Key** position

Clinic (clinicNo, clinicName, clinicAddress, clinicTelephone, manager)

- **Primary Key** clinicNo
- **Alternate Key** clinicAddress, **Alternate Key** clinicName, **Alternate Key** clinicTelephone
- **Foreign Key** manager **references** Staff(staffNo)

Pet (petNo, petDOB, species, breed, color, clinicNo, ownerNo)

- **Primary Key** petNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo)
- **Foreign Key** ownerNo **references** Owner(ownerNo)
- **Foreign Key** species **references** Animal(species)
- **Foreign Key** breed **references** Animal(breed)

Animal (species, breed)

- **Primary Key** species, breed

Owner (ownerNo, ownerFName, ownerLName, ownerAddress, ownerTelephone)

- **Primary Key** ownerNo

Examination (examNo, complaint, description, date, action, petNo, staffNo)

- **Primary Key** examNo

- **Foreign Key** petNo **references** Pet(petNo)
- **Foreign Key** staffNo **references** Staff(staffNo)

c) Validate the logical model against 5 user transactions

1. **Register a new pet:** insert the pet's details into the Pet table and assign a clicNo to the pet. If the pet belongs to a new owner, register the owner’s details in the Owner table. Moreover, ensure that the species and breed exist in the Animal table.
2. **Update a clinic’s manager:** update the manager field in the Clinic table. Ensure the new manager's staffNo exists in the Staff table.
3. **Find how many pets a specific owner has:** count the number of rows in the Pet table where ownerNo matches the given owner.
4. **Find the average salary of each position:** group the rows in the Position table by the position attribute, and calculate the average salary.
5. **Count the number of pets registered in each clinic:** group the rows in the Pet table by the clinicNo attribute, and count the number of rows for each clinic.

d) Define integrity constraints

i. Primary key constraints

Every primary key must be unique, not null, and immutable. Therefore, **staffNo**, **position**, **clinicNo**, **petNo**, **species**-**breed**, **ownerNo**, **examNo** must be not null and unique. Assuming that it’s possible for position to have duplicate values (a cardiologist and radiologist veterinarians could both be labeled as ‘doctor’), we will add a new primary key to the Position relation (**positionNo**) which will act as a unique identifier for each role, and relabel **position** to **positionName** for clarity.

Staff (staffNo, staffFName, staffLName, staffAddress, staffDOB, positionNo, clinicNo)

- **Primary Key** staffNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo)
- **Foreign Key** positionNo **references** Position(positionNo)

Position (positionNo, positionName, salary)

- **Primary Key** positionNo

ii. Referential integrity/Foreign key constraints

Referential integrity means that if the foreign key contains a value, that value must refer to an existing tuple in the parent relation. There are two issues regarding foreign keys that must be addressed. The first considers whether nulls are allowed for the foreign key - that is, if the participation of the child relation in the relationship is **mandatory**, then nulls are not allowed, and if it’s **optional**, then nulls are allowed.

For our logic model, we first will analyze the relations that include FK's, and determine whether they are allowed to be null.

1. Staff (staffNo, staffFName, staffLName, staffAddress, staffDOB, positionNo, clinicNo)

- **Primary Key** staffNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo)
    - Nulls are not allowed for **clinicNo**, since its participation is mandatory (1..1 multiplicity: every staff member must be assigned to minimum one and maximum one clinic).
- **Foreign Key** positionNo **references** Position(positionNo)
    - Nulls are not allowed for **clinicNo**, since its participation is mandatory (1..1 multiplicity: every staff member must be assigned to minimum one and maximum one role).

2. Clinic (clinicNo, clinicName, clinicAddress, clinicTelephone, manager)

- **Primary Key** clinicNo
- **Alternate Key** clinicAddress, **Alternate Key** clinicName, **Alternate Key** clinicTelephone
- **Foreign Key** manager **references** Staff(staffNo)
    - Nulls are not allowed for **manager**, since participation is mandatory (1..1 multiplicity: every clinic must have minimum one and maximum one manager).

3. Pet (petNo, petDOB, species, breed, color, clinicNo, ownerNo)

- **Primary Key** petNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo)
    - Nulls are not allowed for **clinicNo**, since participation is mandatory (1..1 multiplicity: every pet must be registered at minimum one and maximum one clinic).
- **Foreign Key** ownerNo **references** Owner(ownerNo)
    - Nulls are not allowed for **ownerNo**, since participation is mandatory (1..1 multiplicity: every pet must have minimum one and maximum one owner).
- **Foreign Key** species **references** Animal(species)
    - Nulls are not allowed for **species**, since participation is mandatory (1..1 multiplicity: every pet must belong to minimum one and maximum one species).
- **Foreign Key** breed **references** Animal(breed)
    - Nulls are not allowed for **breed**, since participation is mandatory (1..1 multiplicity: every pet must belong to minimum one and maximum one breed. A mixed breed pet will be registered as a new breed).

4. Examination (examNo, complaint, description, date, action, petNo, staffNo)

- **Primary Key** examNo
- **Foreign Key** petNo **references** Pet(petNo)

        - Nulls are not allowed for **petNo**, since participation is mandatory (1..1 multiplicity: every examination must include minimum one and maximum one pet).
    - **Foreign Key** staffNo **references** Staff(staffNo)
        - Nulls are not allowed for **staffNo**, since participation is mandatory (1..1 multiplicity: every examination must be conducted by minimum one and maximum one staff member).

Now that the nullity of foreign keys has been established, we address existence constraints that define conditions under which a candidate key or foreign key may be inserted, updated, or deleted.

1. Staff (<u>staffNo</u>, staffFName, staffLName, staffAddress, staffDOB, positionNo, clinicNo)

- **Primary Key** staffNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo) **ON UPDATE CASCADE ON DELETE NO ACTION**
    - **ON UPDATE CASCADE:** when the primary key (**clinicNo**) of the parent tuple (**Clinic**) is updated, the updates are reflected in any referencing child tuples (**Staff**) in a cascading manner.
    - **ON DELETE NO ACTION:** prevent a deletion from the parent relation if there are any referenced child tuples. You cannot delete a clinic if it currently employs staff members.
- **Foreign Key** positionNo **references** Position(positionNo) **ON UPDATE CASCADE ON DELETE NO ACTION**
    - **ON UPDATE CASCADE:** when the primary key (**positionNo**) of the parent tuple (**Position**) is updated, the updates are reflected in any referencing child tuples (**Staff**) in a cascading manner.
    - **ON DELETE NO ACTION:** prevent a deletion from the parent relation if there are any referenced child tuples. You cannot delete a position if there are currently employees that hold that position.

2. Clinic (<u>clinicNo</u>, clinicName, clinicAddress, clinicTelephone, manager)

- **Primary Key** clinicNo
- **Alternate Key** clinicAddress, **Alternate Key** clinicName, **Alternate Key** clinicTelephone[2]
- **Foreign Key** manager **references** Staff(staffNo) **ON UPDATE CASCADE ON DELETE NO ACTION**
    - **ON UPDATE CASCADE:** when the primary key (**staffNo**) of the parent tuple (**Staff**) is updated, the updates are reflected in any referencing child tuples (**Clinic**) in a cascading manner.

[2] Following the same assumptions made in the conceptual data model: each clinic has a unique name, address, and telephone.

        - **ON DELETE NO ACTION:** prevent a deletion from the parent relation if there are any referenced child tuples. You cannot delete a staff member if they currently manage a clinic. You must first assign a new manager for that clinic, and then delete the former manager.

3. Pet (petNo, petDOB, species, breed, color, clinicNo, ownerNo)

- **Primary Key** petNo
- **Foreign Key** clinicNo **references** Clinic(clinicNo) **ON UPDATE CASCADE ON DELETE NO ACTION**
    - **ON UPDATE CASCADE:** when the primary key (**clinicNo**) of the parent tuple (**Clinic**) is updated, the updates are reflected in any referencing child tuples (**Pet**) in a cascading manner.
    - **ON DELETE NO ACTION:** prevent a deletion from the parent relation if there are any referenced child tuples. You cannot delete a clinic before first reassigning the pets registered in it to another clinic.
- **Foreign Key** ownerNo **references** Owner(ownerNo) **ON UPDATE CASCADE ON DELETE CASCADE**
    - **ON UPDATE CASCADE:** when the primary key (**ownerNo**) of the parent tuple (**Owner**) is updated, the updates are reflected in any referencing child tuples (**Pet**) in a cascading manner.
    - **ON DELETE CASCADE:** when the parent tuple (**Owner**) is deleted, automatically delete any referenced child tuples (**Pet)**. That is, if an owner is deleted, all pets they own are deleted as well, if they haven't been previously re-assigned.
- **Foreign Key** (species, breed) **references** Animal(species, breed) **ON UPDATE CASCADE ON DELETE NO ACTION**
    - **ON UPDATE CASCADE:** when the primary key (**species, breed**) of the parent tuple (**Animal**) is updated, the updates are reflected in any referencing child tuples (**Pet**) in a cascading manner.
    - **ON DELETE NO ACTION:** prevent a deletion from the parent relation if there are any referenced child tuples. You cannot delete an animal if there are pets that are that animal.

4. Examination (examNo, complaint, description, date, action, petNo, staffNo)

- **Primary Key** examNo
- **Foreign Key** petNo **references** Pet(petNo) **ON UPDATE CASCADE ON DELETE CASCADE**
    - **ON UPDATE CASCADE:** when the primary key (**petNo**) of the parent tuple (**Pet**) is updated, the updates are reflected in any referencing child tuples (**Examination**) in a cascading manner.
    - **ON DELETE CASCADE:** when the parent tuple (**Pet**) is deleted, automatically delete any referenced child tuples (**Examination)**. That is, if a pet is deleted, all of their past examinations are deleted as well.

- **Foreign Key** staffNo **references** Staff(staffNo) **ON UPDATE CASCADE ON DELETE SET DEFAULT**
    - **ON UPDATE CASCADE:** when the primary key (**staffNo**) of the parent tuple (**Stadd**) is updated, the updates are reflected in any referencing child tuples (**Examination**) in a cascading manner.
    - **ON DELETE SET DEFAULT:** when the parent tuple (**Staff**) is deleted, the foreign key values in all corresponding child tuples (**Examination**) should automatically be set to their default values. If a staff member is deleted, all the examinations they conducted will be assigned to the clinic manager as a default.

iii. Alternate key constraints (if any)

In the **Clinic** relation, **clinicName**, **clinicAddress**, **clinicTelephone** are all alternate keys for the primary key (**clinicNo**). Consequently, they cannot be null.

iv. Required data & v. Attribute domain constraints.

1. **Staff**: the primary and foreign keys in the Staff relation may not be null, as stated above. Moreover, staff members must have a first name and last name. It is not mandatory to include the staff member's date of birth or address.

Attribute	Required	Domain Constraints
staffNo	NOT NULL	6-character string starting with the letter S (for Staff) and followed by 5 digits (i.e, S12345)
staffFName	NOT NULL	String
staffLName	NOT NULL	String
staffAddress	NULL	String
staffDOB	NULL	Date (MM/DD/YYYY)
positionNo	NOT NULL	6-character string starting with the letter P (for Position) and followed by 5 digits (i.e, P12345)
clinicNo	NOT NULL	6-character string starting with the letter C (for Clinic) and followed by 5 digits (i.e, C12345)

2. **Position**: the primary key in the Position relation may not be null, as stated above. All positions must have a position name and a salary associated with them.

Attribute	Required	Domain Constraints
positionNo	NOT NULL	6-character string starting with the letter J (for Job, since P is taken by Pet) and followed by 5 digits (i.e, J12345)
positionName	NOT NULL	String
salary	NOT NULL	Float with two decimals

3. **Clinic**: the primary and foreign keys in the Clinic relation may not be null, as stated above. All clinics must have a name, address, and telephone.

Attribute	Required	Domain Constraints
clinicNo	NOT NULL	6-character string starting with the letter C (for Clinic) and followed by 5 digits (i.e, C12345)
clinicName	NOT NULL	String
clinicAddress	NOT NULL	String
clinicTelephone	NOT NULL	10-digit integer
manager	NOT NULL	6-character string starting with the letter S (for Staff) and followed by 5 digits (i.e, S12345)

4. **Pet**: the primary and foreign keys in the Pet relation may not be null, as stated above. It is mandatory to include the pet's date of birth, because that can help a doctor make a diagnosis. It is not mandatory to include the pet's color.

Attribute	Required	Domain Constraints
petNo	NOT NULL	6-character string starting with the letter P (for Pet) and followed by 5 digits (i.e, P12345)

petDOB	NOT NULL	Date (MM/DD/YYYY)
species	NOT NULL	String
breed	NOT NULL	String
color	NULL	String
clinicNo	NOT NULL	6-character string starting with the letter C (for Clinic) and followed by 5 digits (i.e, C12345)
ownerNo	NOT NULL	6-character string starting with the letter O (for Owner) and followed by 5 digits (i.e, O12345)

5. **Animal**: the primary keys in the Animal relation may not be null, as stated above.

Attribute	**Required**	**Domain Constraints**
species	NOT NULL	String
breed	NOT NULL	String

6. **Owner**: the primary and foreign keys in the Owner relation may not be null, as stated above. All owners must have a first and last name, and a telephone number. It is not mandatory to store the owner's address.

Attribute	**Required**	**Domain Constraints**
ownerNo	NOT NULL	6-character string starting with the letter O (for Owner) and followed by 5 digits (i.e, O12345)
ownerFName	NOT NULL	String
ownerLName	NOT NULL	String
ownerAddress	NULL	String
ownerTelephone	NOT NULL	10-digit integer

7. **Examination**: the primary and foreign keys in the Examination relation may not be null, as stated above. All examinations must have a complaint, description, date and action. In the case no

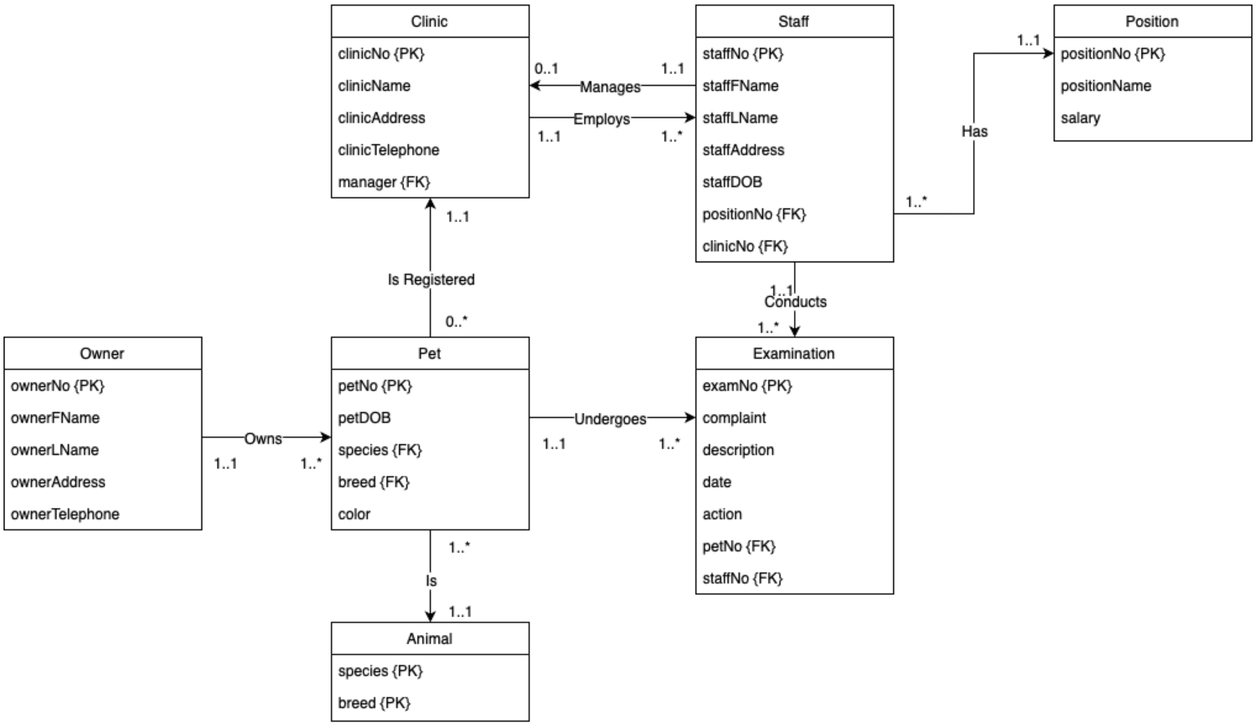
action is taken as a result of the examination, the value would be ‘No action taken.’

Attribute	Required	Domain Constraints
examNo	NOT NULL	6-character string starting with the letter E (for Examination) and followed by 5 digits (i.e, E12345)
complaint	NOT NULL	String
description	NOT NULL	String
date	NOT NULL	Date (MM/DD/YYYY)
action	NOT NULL	String
petNo	NOT NULL	6-character string starting with the letter P (for Pet) and followed by 5 digits (i.e, P12345)
staffNo	NOT NULL	6-character string starting with the letter S (for Staff) and followed by 5 digits (i.e, S12345)

vi. General constraints (if any).

- Only one staff member can manage a clinic.
- Every pet must have an owner.
- Every pet must be at one clinic per examination.
- Every examination must be linked with a pet.
- Every examination must be conducted by a staff member.
- The salary for the staff must be a positive float.
- staffDOB must occur at a date earlier than today.
- petDOB and date (in the Examination relation) must occur either today or at an earlier date.
    - This is to account for the event that a pet is born in the clinic.
    - We are assuming there is no scheduling function that stores information for future examinations in the database.

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


Clinic
clinicNo {PK}
clinicName
clinicAddress
clinicTelephone
manager {FK}
Staff
staffNo {PK}
staffFName
staffLName
staffAddress
staffDOB
positionNo {FK}
clinicNo {FK}
Position
positionNo {PK}
positionName
salary
0..1
Manages
1..1
Employs
1..1
1..*
1..1
Has
1..*
1..1
Is Registered
0..*
1..1
Conducts
1..*
Owner
ownerNo {PK}
ownerFName
ownerLName
ownerAddress
ownerTelephone
Owns
1..1
1..*
Pet
petNo {PK}
petDOB
species {FK}
breed {FK}
color
Undergoes
1..1
1..*
Examination
examNo {PK}
complaint
description
date
action
petNo {FK}
staffNo {FK}
1..*
Is
1..1
Animal
species {PK}
breed {PK}