

# Project - Part 2

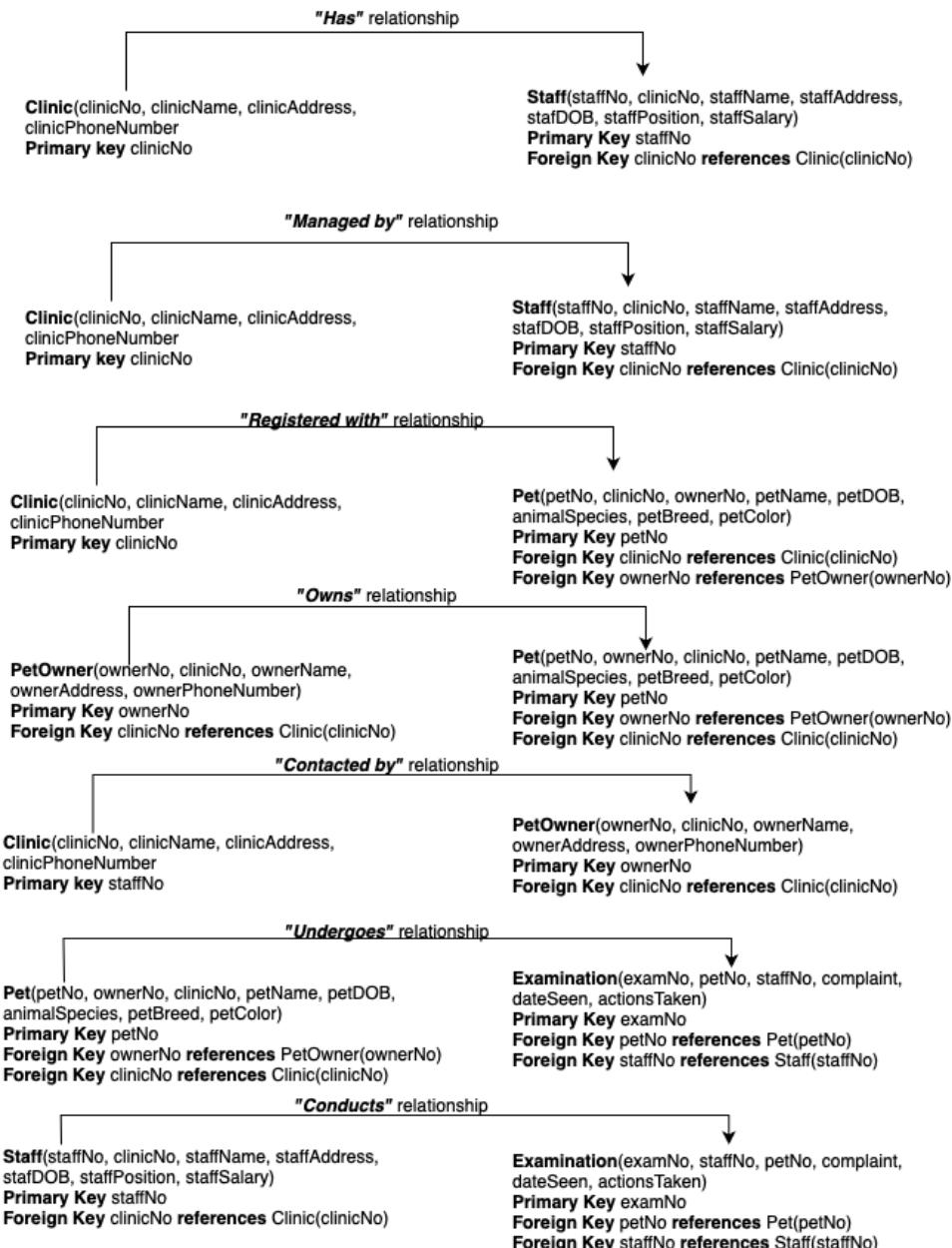
CSC423/CSC623 Database Systems

Natalie Gutierrez, Anna Valente

November 21, 2024

**Develop a logical data model based on the following requirements:**

**a. Derive relations from the conceptual model.**



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

<b>Clinic</b> (clinicNo, clinicName, clinicAddress, clinicPhoneNumber <b>Primary key</b> clinicNo
---

- The “Clinic” relation is in 3NF. All non-primary key attributes have full dependency on the primary key, and there are no partial or transitive dependencies.

<b>Staff</b> (staffNo, clinicNo, staffName, staffAddress, staffDOB, staffPosition, staffSalary) <b>Primary Key</b> staffNo <b>Foreign Key</b> clinicNo <b>references</b> Clinic(clinicNo)
--

- The “Staff” relation is in 3NF. All non-primary key attributes have full dependency on the primary key, including the foreign key. There are no partial or transitive dependencies.

<b>Pet</b> (petNo, clinicNo, ownerNo, petName, petDOB, animalSpecies, petBreed, petColor) <b>Primary Key</b> petNo <b>Foreign Key</b> clinicNo <b>references</b> Clinic(clinicNo) <b>Foreign Key</b> ownerNo <b>references</b> PetOwner(ownerNo)
--

- The “Pet” relation is in 3NF. All non-primary key attributes have full dependency on the primary key, including the foreign keys. There are no partial or transitive dependencies.

<b>PetOwner</b> (ownerNo, clinicNo, ownerName, ownerAddress, ownerPhoneNumber) <b>Primary Key</b> ownerNo <b>Foreign Key</b> clinicNo <b>references</b> Clinic(clinicNo)
---

- The “PetOwner” relation is in 3NF. All non-primary key attributes have full dependency on the primary key, including the foreign keys. There are no partial or transitive dependencies.

<b>Examination</b> (examNo, petNo, staffNo, complaint, dateSeen, actionsTaken) <b>Primary Key</b> examNo <b>Foreign Key</b> petNo <b>references</b> Pet(petNo) <b>Foreign Key</b> staffNo <b>references</b> Staff(staffNo)
--

- The “Examination” relation is in 3NF. All non-primary key attributes have full dependency on the primary key, including the foreign keys. There are no partial or transitive dependencies.

**c. Validate the logical model against 5 user transactions. (Note: These will be then implemented in 3c).**

1. Register a new pet with the Clinic "Healthy Paws" (clinicNo 0011). The pet's name is "Brandy" and was born May 5, 2017; is a dog and a yellow golden retriever; and her owner is Jane Smith (petOwnerNo = 189). Jane Smith's address is 123 Yellow Road, Coral Gables, FL 33146, and her phone number is 305-555-1234.
- All of the details for the pet can be held in and found in the *Pet* entity. All of the details for the pet's owner can be held in and found in the *PetOwner* entity. Assuming that the clinic the pet is registered to is already in the database, we can link the required clinic to the pet and pet owner using the "clinicNo" attribute in the *Clinic* entity. We can use the *Pet* registered with *Clinic*, *PetOwner* owns *Pet*, and *Clinic* contacted by *Owner* relationships for this transaction.
2. An examination (examNo = 0021) is scheduled for Charlie, a black Labrador retriever (petNo = 201), at the clinic "Healthy Paws" (clinicNo = 0011). The examination was conducted by Lisa Martinez (staffNo = 1015) on November 20, 2024. The complaint is "persistent itching," and the description notes "allergic reaction suspected." Actions taken include "skin test and prescribed hypoallergenic shampoo."
- All of the details for the examination can be held in and found in the *Examination* entity. Assuming the clinic, the pet, and the staff member are already in the database, we can link the examination to the pet, the clinic, and the staff member who conducted it using each of their primary keys: examNo, clinicNo, and staffNo. We can use the *Pet* undergoes *Examination*, *Staff* undergoes *Examination*, *Clinic* has *Staff*, and *Pet* registered with *Clinic* relationships for this transaction.
3. List details of the examinations performed by the staff member Lisa Martinez (staffNo = 1015) at the clinic "Healthy Paws" (clinicNo = 0011).
- All of the details for each examination are held in the *Examination* entity. Assuming Lisa Martinez and the clinic "Healthy Paws" are in the database, we can use the primary key staffNo from the *Staff* entity and the foreign key clinicNo in the *Staff* entity to fetch all of

the examinations she's done in the *Examination* entity. We can use the *Staff* conducts *Examination* relationship and *Clinic* has *Staff* relationship for this transaction.

4. Fetch all of the staff's name and position, and the staff manager's name, position, and salary, for the clinic "Healthy Paws" (clinicNo = 0011).
- All of the details for each member of the staff, including their name, position, and salary are in the *Staff* entity. Assuming the clinic "Healthy Paws" is already in the database, we can use the clinicNo foreign key in the *Staff* entity, and the staffName, staffPosition, and staffSalary attributes in the *Staff* entity to retain the information needed. For the regular staff, we can use the *Clinic* has *Staff* relationship for this transaction, and for the staff manager, we can use the *Clinic* managed by *Staff* relationship for this transaction.
5. Henry Green (staffNo = 1002), currently a "Veterinarian" that makes \$130,000 a year, was born April 5, 1983. He was promoted to Clinic Manager for "Healthy Paws" (clinicNo = 0011) on November 22, 2024. His official updated position is "Veterinarian and Clinic Manager," with a salary adjustment to \$150,000 per year.
- All of the details needed to update Henry Green's position and salary, assuming he is already in the database, are in the *Staff* entity. We have the primary key staffNo in the *Staff* entity to uniquely fetch and update Henry Green's information. We also have the foreign key clinicNo that points back to the *Clinic* relation needed to track which clinic he works at. Since we have all of the details needed to update his information, and we can use the *Clinic* has *Staff* relation, we can complete this transaction.

**d. Define integrity constraints:**

**i. Primary key constraints.**

- Ensure all primary key attributes (clinicNo, staffNo, ownerNo, petNo, examNo) are unique. According to entity integrity, all of these primary keys must not contain any null values.

**ii. Referential integrity/Foreign key constraints.**

- Staff.clinicNo references Clinic.clinicNo.
- Pet.clinicNo references Clinic.clinicNo.
- Pet.ownerNo references PetOwner.ownerNo.
- PetOwner.clinicNo references Clinic.clinicNo.

- Examination.petNo references Pet.petNo.
- Examination.staffNo references Staff.staffNo.
- If participation of the child relation in the relationship is not mandatory, then null values for these foreign keys are allowed; if it is mandatory, then null values for these foreign keys are not allowed.

**iii. Alternate key constraints (if any).**

- Ensure certain attributes like clinicPhoneNumber or ownerPhoneNumber are unique if they act as alternate keys.

**iv. Required data.**

- Non-null constraints on essential fields (e.g. primary keys).
- Every Staff member is required to have a name, hold a position, and have a salary.
- Every Clinic must have a name, address, and phone number.
- Every pet must have an owner attached to it, its name, species, DOB, and breed.
- Every pet owner must have a name and phone number.
- Every examination must document the pet undergone, the staff member who conducted it, the complaint, the date seen, and the actions taken.

**v. Attribute domain constraints.**

- Define valid data types and ranges, such as:

Attribute	Parent Entity	Attribute Domain Constraints
clinicNo	Clinic	Integer, not null
clinicName	Clinic	varchar, max length 40, not null
clinicAddress	Clinic	Varchar, max length 50, Not null
clinicPhoneNumber	Clinic	Varchar, max length 11, not null
staffNo	Staff	Integer, not null
staffName	Staff	Varchar, max length 30, not null
staffAddress	Staff	Varchar, max length 50

staffDOB	Staff	Date data, format: MM-DD-YYYY
staffPosition	Staff	Varchar, max length 30, not null
staffSalary	Staff	Integer, not null
ownerNo	PetOwner	Integer, not null
ownerName	PetOwner	Varchar, max length 30, not null
ownerAddress	PetOwner	Varchar, max length 50
ownerPhoneNumber	PetOwner	Varchar, max length 11, not null
petNo	Pet	Integer, not null
petName	Pet	Varchar, max length 30, not null
petDOB	Pet	Date data, format: MM-DD-YYYY, not null
animalSpecies	Pet	Varchar, max length 30, not null
petBreed	Pet	Varchar, max length 30, not null
petColor	Pet	Varchar, max length 30, not null
examNo	Examination	Integer, not null
complaint	Examination	Varchar, max length 200, not null
dateSeen	Examination	Date data, format: MM-DD-YYYY, not null
actionsTaken	Examination	Varchar, max length 200, not null

vi. General constraints (if any).

- A staff member cannot manage more than one clinic simultaneously (*Clinic* managed by *Staff* relationship constraint). A staff member is not required to manage a clinic.
- A clinic must have at least one staff member (*Clinic* has *Staff* relationship constraint).
- An owner can have one or more pets, but a pet can only belong to one pet owner. A pet owner must own at least one pet (*PetOwner* owns *Pet* relationship constraint).
- A pet can be only registered with one clinic. A clinic can have many pets registered with it, however (*Pet* registered with *Clinic* relationship constraint).
- A pet can undergo 0 or many examinations, but each examination can be registered with only one pet (*Pet* undergoes *Examination* relationship constraint).
- A staff member is not required to do an examination, but each examination must have a staff member registered with it who conducted it (*Staff* conducts *Examination* relationship constraint).

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

