Stephanie Peralta

Dr. Vanessa Aguiar-Pulido

CSC 423 Database Systems

November 17th, 2022

Pawsome Pets: Design, Development, and Implementation of a Relational Database Pt. 2

I. INTRODUCTION

As part of the University of Miami's Fall 2022 CSC 423 Database Systems course with Dr. Vanessa Aguiar-Pulido, students are tasked with designing and creating a relational database for the case study provided:

Case Study: Pawsome Pets

A company called Pawsome Pets runs multiple clinics. The company would like for their data to be stored in a database. The following description was obtained during the analysis phase:

"Each of the Pawsome Pets clinics has several staff members and a member of staff manages at most one clinic (not all staff manage clinics). Each clinic has a unique clinic number (clinicNo) and each member of staff has a unique staff number (staffNo). Additionally, the company would like to store each clinic's name, address and telephone number, as well as the staff's name, address, telephone number, DOB, position and salary.

When a pet owner contacts a clinic, the owner's pet is registered with the clinic. An owner can own one or more pets, but a pet can only be registered at one clinic. Each owner has a unique owner number (ownerNo), a name, an address, and a telephone number. Each pet has a unique pet number (petNo), name, DOB, animal species, breed, and color.

When the pet comes to the clinic, it undergoes an examination by a member of the consulting staff. The database should store the following information for each examination: chief complaint (i.e., the main cause for the visit), description (i.e., what was done during the examination), date seen, and actions taken (e.g., a treatment was prescribed, tests were ordered). A unique examination number (examNo) is assigned to each examination."

The project is divided into three parts, each constituting a piece of the overall design, development, and implementation process for creating the relational database. The following report is the second part of this project: developing a logical data model of the case study.

II. LOGICAL DATA MODEL DEVELOPMENT

The conceptual model developed in the first report must be translated into a logical model to ensure correct relationships between each entity. This is done through the following steps:

a. Derive relations from the conceptual model.

Relations are derived based on the kind of structures found in the model. In this case, the structures that occur and the manner in which they are handled are:

a. Strong entity types – relations containing all simple attributes of those entities are created. Note that the following assumption is made: each clinic, staff member, and pet owner has a unique telephone number.

i. **Clinic** {clinicNo, name, address, telephoneNumber}

Primary Key clinicNo

Alternate Key telephoneNumber

ii. **Staff** {staffNo, name, address, telephoneNumber, DOB, position, salary}

Primary Key staffNo

Alternate Key telephoneNumber

iii. **Owner** {ownerNo, name, address, telephoneNumber}

Primary Key ownerNo

Alternate Key telephoneNumber

iv. **Pet** {petNo, name, DOB, species, breed, color}

Primary Key petNo

v. **Examination** {examNo, chiefComplaint, description, dateSeen, actionsTaken}

Primary Key examNo

b. One-to-many relationship types – primary keys of "parent" entities are copied into "child" entity relations to act as foreign keys.

i. **Clinic** *Employs* **Staff** – one clinic employs many staff members. Then,

Staff {staffNo, name, address, telephoneNumber, DOB, position, salary, clinicNo}

Primary Key staffNo

Alternate Key telephoneNumber

Foreign Key clinicNo **references** Clinic(clinicNo)

ii. **Clinic** *Registers* **Pet** – one clinic has many pets registered. Then,

Pet {petNo, name, DOB, species, breed, color, clinicNo}

Primary Key petNo

Foreign Key clinicNo **references** Clinic(clinicNo)

iii. **Owner** *Owns* **Pet** – one owner can own many pets. Then,

Pet {petNo, name, DOB, species, breed, color, clinicNo, ownerNo}

Primary Key petNo

Foreign Key clinicNo **references** Clinic(clinicNo)

Foreign Key ownerNo **references** Owner(ownerNo)

iv. **Staff** *Performs* **Examination** – one staff member can perform multiple examinations. Then,

Examination {examNo, chiefComplaint, description, dateSeen, actionsTaken, staffNo}

Primary Key examNo

Foreign Key staffNo **references** Staff(staffNo)

v. **Pet** *Undergoes* **Examination** – one pet can undergo many examinations. Then,

Examination {examNo, chiefComplaint, description, dateSeen, actionsTaken, staffNo, petNo}

Primary Key examNo

Foreign Key staffNo **references** Staff(staffNo)

Foreign Key petNo **references** Pet(petNo)

c. One-to-one relationship types – depend on participation constraints.

Mandatory participation on one side – entity with optional constraint is the parent and entity with mandatory constraint is the child. Primary keys of parent entities are copied into child entity relations to act as foreign keys.

i. **Staff** *Manages* **Clinic** – every clinic has a manager but not all staff are managers. **Staff** is the parent entity and **Clinic** is the child entity. Then,

Clinic {clinicNo, name, address, telephoneNumber, staffNo}

Primary Key clinicNo

Alternate Key telephoneNumber

Foreign Key staffNo **references** Staff(staffNo)

b. Validate the logical model using normalization to Third Normal Form (3NF).

3NF eliminates any partial and functional dependencies. Each relation's dependencies are thus checked to ensure only primary and candidate key dependencies exist. Note that the relations have been written in schema notation and the primary key of each has been underlined.

a. **Clinic** {<u>clinicNo</u>, name, address, telephoneNumber, staffNo}

The functional dependencies found in this relation are:

Primary key: clinicNo → name, address, telephoneNumber, staffNo

Candidate key: telephoneNumber → name

b. **Staff** {<u>staffNo</u>, name, address, telephoneNumber, DOB, position, salary, clinicNo}

The functional dependencies found in this relation are:

Primary key: staffNo → name, address, telephoneNumber, DOB, position, salary, clinicNo

Candidate key: telephoneNumber → name

c. **Owner** {<u>ownerNo</u>, name, address, telephoneNumber}

The functional dependencies found in this relation are:

Primary key: ownerNo → name, address, telephoneNumber

Candidate key: telephoneNumber → name

d. **Pet** {petNo, name, DOB, species, breed, color, clinicNo, ownerNo}

The functional dependencies found in this relation are:

Primary key: petNo → name, DOB, species, breed, color, clinicNo, ownerNo

e. **Examination** {examNo, chiefComplaint, description, dateSeen, actionsTaken, staffNo, petNo}

The functional dependencies found in this relation are:

Primary key: examNo → chiefComplaint, description, dateSeen, actionsTaken, staffNo, petNo

c. Validate the logical model against user transactions.

To further validate the derived relations, potential questions of user transactions can be formulated with the steps to their solutions outlined. Note that any specific names or numbers mentioned have been created for the sake of formulating the question and will be included in the implementation of the database.

a. "Find how many pets owner number JJ225 owns."

The **Owner** and **Pet** relations are related via the foreign key Pet(ownerNo), through which they are joined. Then owner number JJ225 is found, and the number of pets they own is counted and printed.

b. "Find how many dogs are registered in clinic number NY102."

The **Clinic** and **Pet** relations are related via the foreign key Pet(clinicNo), through which they are joined. Then clinic number NY102 is found, pets are filtered by species to select only dogs, and the number of dogs is counted and printed.

c. “Find the average salary of the staff in Big Apple Pet Clinic.”

   The **Clinic** and **Staff** relations are related via the foreign key Staff(clinicNo), through which they are joined. Then Big Apple Pet Clinic is found, and the average salary of its staff is calculated and printed.

d. “List the names and species of the pets that have had annual checkups in 2022.”

   The **Pet** and **Examination** relations are related via the foreign key Examination(petNo), through which they are joined. Then the examinations are filtered by the chief complaint to select only annuals and the date to select only those exams done in 2022, and the required information is printed.

e. “List the names and positions of the staff members who have performed X-Rays and the actions taken at the end of those examinations.”

   The **Staff** and **Examination** relations are related via the foreign key Examination(staffNo), through which they are joined. Then the examinations are filtered by description to select only those where X-Rays were done, and the required information is printed.

d. Define integrity constraints.

The first constraint to consider is that of primary keys. This enforces the principle of entity integrity, which states that the primary key of a relation must have a unique, nonnull value for each row. Thus, Clinic(clinicNo), Staff(staffNo), Owner(ownerNo), Pet(petNo), and Examination(examNo) must contain unique, nonnull values. This will be done in the implementation.

Next is foreign key constraints which enforce referential integrity, the principle that if a foreign key contains a value, that value must refer to an existing, valid row in the parent relation. This can be handled by specifying the action that should be taken when trying to update or delete any values relating to the foreign key.

a. **Clinic** {clinicNo, name, address, telephoneNumber, staffNo}

   Primary Key clinicNo

   Alternate Key telephoneNumber

   Foreign Key staffNo **references** Staff(staffNo) ON UPDATE CASCADE ON DELETE CASCADE

b. **Staff** {staffNo, name, address, telephoneNumber, DOB, position, salary, clinicNo}

   Primary Key staffNo

   Alternate Key telephoneNumber

   Foreign Key clinicNo **references** Clinic(clinicNo) ON UPDATE CASCADE

c. **Pet** {petNo, name, DOB, species, breed, color, clinicNo, ownerNo}

Primary Key petNo

Foreign Key clinicNo **references** Clinic(clinicNo) ON UPDATE CASCADE

Foreign Key ownerNo **references** Owner(ownerNo) ON UPDATE CASCADE ON DELETE SET NULL

d. **Examination** {examNo, chiefComplaint, description, dateSeen, actionsTaken, staffNo, petNo}

   Primary Key examNo

   Foreign Key staffNo **references** Staff(staffNo) ON UPDATE CASCADE

   Foreign Key petNo **references** Pet(petNo) ON UPDATE CASCADE

Alternate key constraints must also be considered and would follow the same rules as primary key constraints. Thus, Clinic(telephoneNumber), Staff(telephoneNumber), and Owner(telephoneNumber) must contain unique, nonnull values. This will be done in the implementation.

There are also constraints for required data, which tells which attributes cannot contain null values.

e. For **Clinic**, no attributes can contain null values.
f. For **Staff**, no attributes can contain null values.
g. For **Owner**, no attributes can contain null values.
h. For **Pet**, no attributes except {name} can contain null values. This is assuming the case where an owner may not have named their pet yet at the time of

registration because it is new or recently acquired; this would then be updated once the pet is named.

i. For **Examination**, no attributes except {actionsTaken} can contain null values as not all examinations may lead to post-exam actions being taken, like the prescription of medication.

Attribute domain constraints must also be specified, which determine the set of legal values attributes can contain. They influence each attribute's data type.

j. For **Clinic**,

{clinicNo} must begin with two characters, followed by three integers.

{name} must be a character string.

{address} must contain characters and integers.

{telephoneNumber} must be a valid 10-digit number.

k. For **Staff**,

{staffNo} must begin with a character, followed by three integers.

{name} and {position} must be character strings.

{address} must contain characters and integers.

{telephoneNumber} must be a valid 10-digit number.

{DOB} must be a valid date in YYYY-MM-DD format.

{salary} must contain integers.

l. For **Owner**,

{ownerNo} must begin with two characters, followed by three integers.

{name} must be a character string.

{address} must contain characters and integers.

{telephoneNumber} must be a valid 10-digit number.

m. For **Pet**,

{petNo} must begin with a character, followed by three integers.

{name}, {species}, {breed}, and {color} must be character strings.

{DOB} must be a valid date in YYYY-MM-DD format.

n. For **Examination**,

{examNo} must begin with a character, followed by three integers.

{chiefComplaint}, {description}, and {actionsTaken} must be character strings.

{dateSeen} must be a valid date in YYYY-MM-DD format.

General constraints can be specified to ensure valid data is entered for each attribute and its integrity is maintained.

o. For **Staff**, {DOB} must not exceed the current date and must be within reasonable time from the current year.

p. For **Pet**, {DOB} must not exceed the current date and must be within reasonable time from the current year.

q. For **Examination**, {dateSeen} must not exceed the current date.

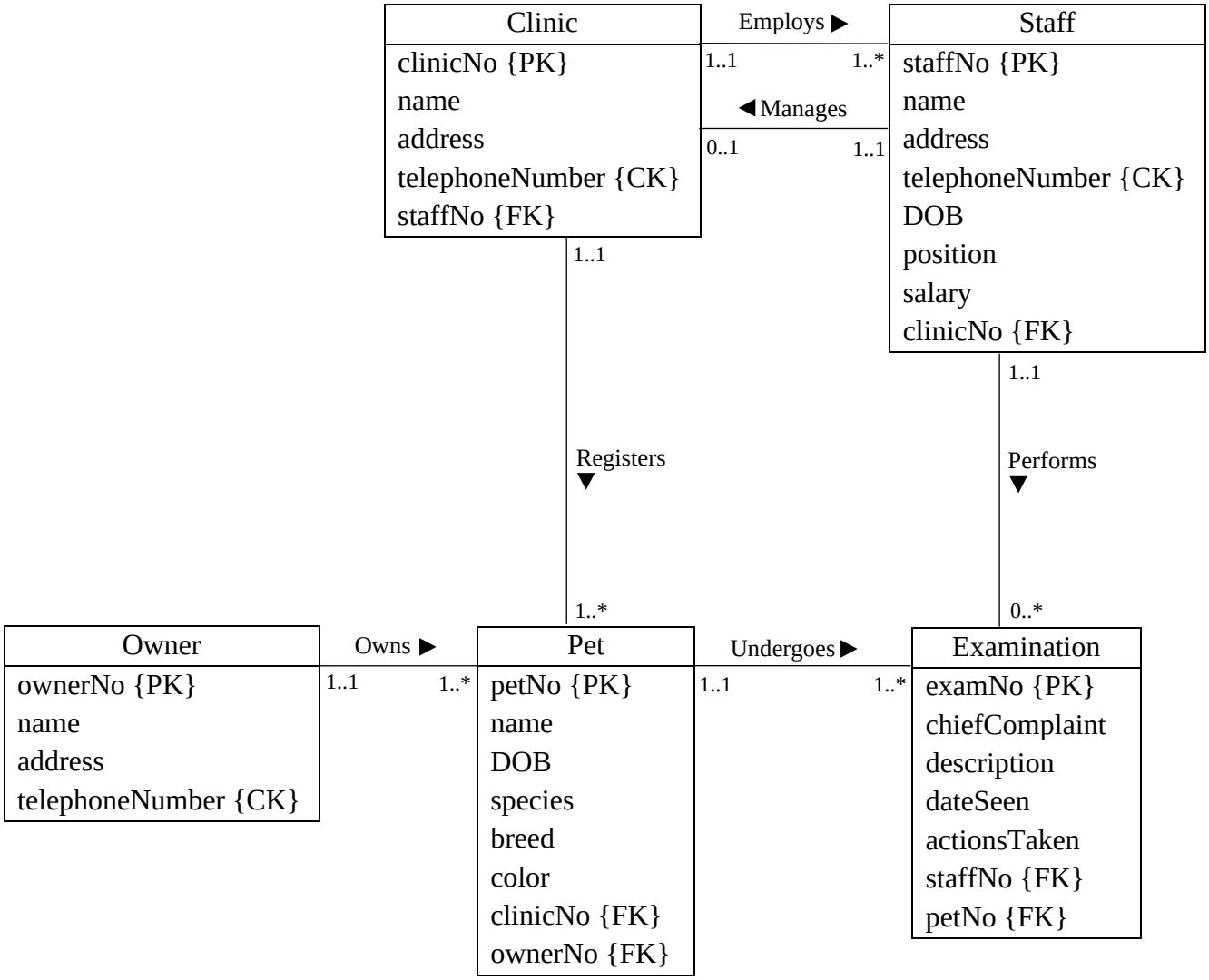
e. Generate the E-R diagram for the logical level (contains FKs as attributes).

III. E-R DIAGRAM

In short, the relations derived from the conceptual model are as follows:

Clinic {clinicNo, name, address, telephoneNumber, staffNo} **Primary Key** clinicNo **Alternate Key** telephoneNumber **Foreign Key** staffNo **references** Staff(staffNo)	**Pet** {petNo, name, DOB, species, breed, color, clinicNo, ownerNo} **Primary Key** petNo **Foreign Key** clinicNo **references** Clinic(clinicNo) **Foreign Key** ownerNo **references** Owner(ownerNo)
Staff {staffNo, name, address, telephoneNumber, DOB, position, salary, clinicNo} **Primary Key** staffNo **Alternate Key** telephoneNumber **Foreign Key** clinicNo **references** Clinic(clinicNo)	**Examination** {examNo, chiefComplaint, description, dateSeen, actionsTaken, staffNo, petNo} **Primary Key** examNo **Foreign Key** staffNo **references** Staff(staffNo) **Foreign Key** petNo **references** Pet(petNo)
Owner {ownerNo, name, address, telephoneNumber} **Primary Key** ownerNo **Alternate Key** telephoneNumber	

With this information, the logical E-R diagram for the Pawsome Pets case study is created, illustrating the details outlined in the previous section.


Clinic
clinicNo {PK}
name
address
telephoneNumber {CK}
staffNo {FK}
Employs ▶
1..1
1..*
◀ Manages
0..1
1..1
Staff
staffNo {PK}
name
address
telephoneNumber {CK}
DOB
position
salary
clinicNo {FK}
1..1
Registers
▼
1..*
1..1
Performs
▼
0..*
Owner
ownerNo {PK}
name
address
telephoneNumber {CK}
Owns ▶
1..1
1..*
Pet
petNo {PK}
name
DOB
species
breed
color
clinicNo {FK}
ownerNo {FK}
Undergoes ▶
1..1
1..*
Examination
examNo {PK}
chiefComplaint
description
dateSeen
actionsTaken
staffNo {FK}
petNo {FK}