

Develop a logical data model based on the following requirements: (11/17/22)

- a) Derive relations from the conceptual model.
  - Staff(staffNo, name, address, phoneNo, DOB, position, salary, clinicNo)
  - Clinic(clinicNo, name, address, phoneNo)
  - Pet(petNo, name, DOB, species, breed, color, ownerNo, clinicNo)
  - Owner(ownerNo, name, address, phoneNo)
  - Examination(examNo, chiefComplaint, description, dateSeen, actions, staffNo, petNo)

<b>Staff</b> (staffNo, name, address, phoneNo, DOB, position, salary) <b>Primary Key</b> ownerNo <b>Foreign Key</b> clinicNo <b>references</b> Cliic(clinicNo)	<b>Clinic</b> (clinicNo, name, address, phoneNo, managerNo) <b>Primary Key</b> clinicNo <b>Foreign Key</b> managerNo <b>references</b> Staff(staffNo) <b>Alternate Key</b> phoneNo <b>Aternate Key</b> address
<b>Pet</b> (petNo, name, DOB, species, breed, color) <b>Primary Key</b> ownerNo <b>Foreign Key</b> clinicNo <b>references</b> Clinic(clinicNo) <b>Foreign Key</b> ownerNo <b>references</b> Owner(ownerNo)	<b>Owner</b> (ownerNo, name, address, phoneNo) <b>Primary Key</b> ownerNo
<b>Examination</b> (examNo, chiefComplaint, description, dateSeen, actions) <b>Primary Key</b> ownerNo <b>Foreign Key</b> staffNo <b>references</b> Staff(staffNo) <b>Foreign Key</b> petNo <b>references</b> Pet(petNo)	

- b) Validate the logical model using normalization to 3NF.
  - Functional Dependencies
    - Staff  
staffNo → name, address, phoneNo, DOB, position, salary, clinicNo (PK)
    - Clinic  
clinicNo → name, address, phoneNo (PK)
    - Pet  
petNo → name, DOB, species, breed, color, ownerNo, clinicNo (PK)
    - Owner  
ownerNo → name, address, phoneNo (PK)
    - Examination  
examNo → chiefComplaint, description, dateSeen, actions, staffNo, petNo (PK)
  - Since there are no partial or transitive dependencies, the model is in 3NF

c) Validate the logical model against user transactions.

- List all clinics and all pets registered there.
  - You would join the Pet and Clinic tables with condition `Pet.clinicNo = Clinic.clinicNo`
- List all the pets of owner O11
  - You would check `Pet.owner = 'O11'` by using the ownerNo foreign key in the Pet relation that references Owner to list all of the pets owned by that owner.
- List all examinations that were performed by the staff member whose staffNo is S05.
  - You would select all attributes from the Examination relation and join the Staff relation using the foreign key staffNo in Examination, referencing Staff where `Examination.staffNo = 'S05'`.
- List all examinations done on pets whose breed is "Havanese" and were done at the clinic whose clinicNo is C2
  - You would join the Examination, Pet, and Staff tables and would list examNo, petNo, breed, staffNo, clinicNo where `Examination.staffNo = Staff.staffNo, Examination.petNo = Pet.petNo, Pet.breed = "Havanese"` and `Staff.clinicNo = 'C2'`.
- List all staff who are managers and have performed more than 5 examinations
  - You would join the Staff and Examinations table and list staffNo, count(examNo) where `Examination.staffNo = Staff.staffNo` and count examNo as examsPerformed grouped by staffNo and check `examsPerformed > 5`.

d) Define integrity constraints:

I. Primary key constraints:

Primary keys examNo, petNo, clinicNo, staffNo, and examNo can not be null by the entity integrity constraint saying no Primary Keys can be null. They must also be unique.

II. Referential integrity/Foreign key constraints:

All Foreign keys must reference an existing value in its parent relation.

III. Alternate key constraints (if any):

Alternate keys address and phoneNo in Clinic relation must be unique and can not be null.

IV. Required data v. Attribute domain constraints.

For required data, all non null attributes must contain valid values.

Attribute Domains

Clinic:

clinicNo: string, name: string, address: string, phoneNo: int, managerNo: string

Staff:

staffNo: string, name: string, address: string, phoneNo: int, DOB: string, position: string, salary: string

Pet:

petNo: string, name: string, DOB: string, species: string, breed: string, color: string

Owner:

ownerNo: string, name: string, address: string, phoneNo: int

Examination:

examNo: string, chiefComplaint: string, description: string, dateSeen: string, actions: string

An attribute domain constraint is that DOB must be in the format “MM-DD-YYYY”.

V. General constraints (if any).

Pets can only belong to one owner.

Staff can only work at one clinic.

Pets can only be registered at one clinic.

Exams can only be performed by one staff member.

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

