**Ch4-5-Review**

**In-Class:** Chapters 4 and 5 Review - In-Class \_\_\_ \_\_\_ \_\_

The following database design shows the relations and relationships in the Veterinary database:

* OWNER (OwnerLastName, OwnerFirstName, OwnerPhone, OwnerEmail)
* PET (PetName, PetType, PetBreed, PetDOB, *OwnerPhone*)
* OwnerPhone in PET must exist in OwnerPhone in Owner.

**Problem 1:** Draw an Entity-Relationship (E-R) Crow's-Foot diagram to show the data model of the Veterinary database.

* Use strong and/or weak entities as appropriate.
* Use non-identifying or identifying relationships as appropriate.
* Specify minimum and maximum cardinalities.

**Problem 2:** Add a CAGE entity to the data model.

* Pets are kept in cages if they are scheduled for or recovering from surgery.
* Each cage is either large, medium, or small.
* Only one pet may be kept in each cage.

**Problem 3:** Update the database design shown above to include a CAGE table.

* Write the definitions of the tables using the format TABLE (Column1, Column2, ...).
* Underline the primary key in each table, and italicize the foreign keys.
* Write the referential integrity check for each foreign key.

OWNER ( OwnerLastName, OwnerFirstName, OwnerPhone, OwnerEmail )

PET ( PetName, PetType, PetBreed, PetDOB, *OwnerPhone* )

CAGE ( CageID, Size, *PetName* )

OwnerPhone in PET must exist in OwnerPhone in OWNER.

PetName in CAGE must exist in PetName in PET.

**Problem 4:** Determine if the new CAGE table needs to be normalized or not.

* Show the candidate key(s).
* Show the functional dependency(s).
* Decide if the relation is well formed or not, and normalize it if needed.

Step 1: Candidate Key(s): CageID

Step 2: Functional Dependencies: CageID 🡪 Size, PetName

Step 3: Are any determinants of the functional dependencies **not** candidate keys? No. So the table is well-formed.

**Problem 5:** Write SQL queries to display the following information from the updated Veterinary database:

* Part a: Display only the pet and owner information.

SELECT \*

FROM OWNER, PET

WHERE PET.OwnerPhone = PET.OwnerPhone;

* Part b: Display only the pet and cage information.

SELECT \*

FROM PET, CAGE

WHERE CAGE.PetName = PET.PetName;

* Part c: Display the pet, owner, and cage information.

SELECT \*

FROM PET, OWNER, CAGE

WHERE OWNER.OwnerPhone = PET.OwnerPhone

AND CAGE.PetName = PET.PetName;