Lee Moriarity

Professor Kelly

CIS 245 ONL01

01 October 2022

Lab #4

PET\_OWNER (OwnerID, OwnerLastName,OwnerFirstName, OwnerPhone, OwnerEmail)

PET (PetID, PetName, PetType, PetBreed, PetDOB, OwnerID)

Questions 3.7-3.29:

3.7 Write an SQL CREATE TABLE statement to create the PET\_OWNER table, with OwnerID as a surrogate key. Justify your choices of column properties. If you are using an actua1 DBMS, also insert the data using SQL.

**CREATE** **TABLE** PET\_OWNER(

OwnerID **int** **NOT** **NULL** **PRIMARY** **KEY**,

OwnerLastName **VARCHAR**(25) **NOT** **NULL**,

OwnerFirstName **VARCHAR**(25) **NOT** **NULL**,

OwnerPhone **VARCHAR**(12) **NULL**,

OwnerEmail **VARCHAR**(100) **NULL**,

);

3.8 Write an SQL CREATE TABLE statement to create the PET table without a referential integrity constraint on OwnerID in PET. Justify your choices of column properties. Why not make every column NOT NULL? If you are using an actual DBMS, also insert the data using SQL.

**CREATE** **TABLE** PET (

PetID **VARCHAR**(20) **PRIMARY** **KEY**,

PetName VARCHAR(20),

PetType **VARCHAR**(20) **NOT** **NULL**,

PetBreed **VARCHAR**(20) **NOT** **NULL**,

PetDOB **DATE**,

OwnerID **VARCHAR**(15)

);

3.9 Create a referential integrity constraint on OwnerID in PET. Assume that deletions should not cascade.

**ALTER** **TABLE** PET **ADD** **CONSTRAINT** reference **FOREIGN** **KEY**(OwnerID) **REFERENCES** PET\_OWNER(OWNERID);

3.10 Create a referential integrity constraint on OwnerID in PET. Assume that deletions should cascade.

**ALTER** **TABLE** PET **ADD** **CONSTRAINT** reference **FOREIGN** **KEY**(OwnerID) **REFERENCES** PET\_OWNER(OWNERID) **ON** **DELETE** **CASCADE**;

3.11 Write the required SQL statements to create the PET\_2 table.

PET\_2 (PetName, PetType, PetBreed, PetDOB, OwnerID)

**CREATE** **TABLE** PET\_2 (

PetName **VARCHAR**(20),

PetType **VARCHAR**(20) **NOT** **NULL**,

PetBreed **VARCHAR**(20) **NOT** **NULL**,

PetDOB **DATE**,

OwnerID **VARCHAR**(15)

);

3.12 Is PET or PET\_2 a better design? Explain your rationale

PET is a better design because it includes PetID.

3.13 Write the SQL statements necessary to remove the PET\_OWNER table from the database. Assume that the referential integrity constraint also needs to be removed. Do not run these commands on the actual database.

**ALTER** **TABLE** PET

**DROP** **CONSTRAINT** OWNER\_KEY;

**DROP** **TABLE** PET\_OWNER;

3.14 Write the SQL statements necessary to remove the PET\_OWNER table from the database.Assume that the PET table also needs to be removed. Do not run these commands on the actual database.

**DROP** **TABLE** PET;

**DROP** **TABLE** PET\_OWNER;

3.15 Write an SQL statement to display all columns of all rows of PET. Do not use the asterick (\*) notation.

**SELECT** PetID, PetName, PetType, PetBreed, PetDOB, OwnerID

**FROM** PET;

3.16 Write an SQL statement to display all columns of all rows of PET. Use the asterick (\*) notation.

**SELECT** **\***

**FROM** PET;

3.17 Write an SQL statement to display the breed and type of all pets.

**SELECT** PetBreed, PetType

**FROM** PET;

3.18 Write an SQL statement to display the breed, type, and DOB of all pets having the type Dog.

**SELECT** PetBreed, PetType, PetDOB

**FROM** PET

**WHERE** PetType **=** 'Dog';

3.19 Write an SQL statement to display the PetBreed column of PET.

**SELECT** PetBreed

**FROM** PET;

3.20 Write an SQL statement to display the PetBreed column of PET. Do not show duplicates

**SELECT** **DISTINCT** PetBreed

**FROM** PET;

3.21 Write an SQL statement to display the breed, type, and DOB for all pets having the type Dog and the breed Std. Poodle.

**SELECT** PetBreed, PetType, PetDOB

**FROM** PET

**WHERE** PetBreed **=** 'Std. Poodle';

3.22 Write an SQL statement to display the name, breed, and type for all pets that are not of type Cat, Dog, Fish

**SELECT** PetName, PetBreed, PetType,

**FROM** PET

**WHERE** PetType **NOT** **IN** ('Dog', 'Bird', 'Fish');

3.23 Write an SWL statement to display the pet ID, breed, and type for all pets having a four-character name starting with K. Note that the RTRIM function will be needed in the solution that uses a CHAR column, but not for one that uses a VARCHAR column.

**SELECT** PetID, PetBreed, PetType

**FROM** PET

**WHERE** PetName **Like** ‘K\_\_\_’);

**SELECT** PetID, PetBreed, PetType

**FROM** PET

**WHERE** RTRIM(PetName **Like** 'K\_\_\_'));

3.24 Write an SQL statement to display the last name, first name, and email of all owners who have an email address ending with somewhere.com. Assume that email account names can be any number of characters. Note that the RTRIM function will be needed in the solution that uses a CHAR column, but not for one that uses a VARCHAR column.

**SELECT** OwnerLastName, OwnerFirstName, OwnerEmail

**FROM** PET\_OWNER

**WHERE** OwnerEmail **LIKE** '%something.com';

**SELECT** RTRIM(OwnerLastName, OwnerFirstName, OwnerEmail

**FROM** PET\_OWNER

**WHERE** RTRIM(OwnerEmail **LIKE** '%something.com');

3.25 Write an SQL statement to display the last name, first name, and email of any owner who has a NULL value for OwnerPhone.

**SELECT** OwerLastName, OwnerFirstName, OwnerEmail

**FROM** PET\_OWNER

**WHERE** OwnerPhone **IS** **NULL**;

3.26 Write an SQL statement to display the name and breed of all pets, sorted by PetName.

**SELECT** PetName, PetBreed

**FROM** PET

**ORDER** **BY** PetName;

3.27 Write an SQL statement to display the breed, type, and DOB.

**SELECT** PetBreed, PetType, PetDOB

**FROM** PET;

3.28 Write an SQL statement to count the number of pets.

**SELECT** **Count**(PedID)

**FROM** PET;

2.39 Write an SQL statement to count the number of distinct breeds.

**SELECT** **Count**(**DISTINCT** PetBreed)

**FROM** PET;

Graphical user interface, text, application, email

Description automatically generated