

Entity relationship data model and normalization computer science



**ASSIGN
BUSTER**

Draw an Entity Relationship Data Model that describes the content and structure of the data held by Petcare. Specify the cardinality ratio and participation constraint of each relationship type.

Answer: Draw an Entity Relationship Data Model

To design the database system for Petcare, we need to draw an Entity Relationship Diagram to show all entities and the relationship between them. The ERD models the static relationships amongst this information that are preserved and maintained by the database. The ERD shows how items of data relate, statically, to each other.

1. 1. 1 Entities of the ERD

First step, we need to define the entities in the database.

According to the requirement of the Petcare, the below entities should be involved in the database system to record the Petcare's information;

- Information of the pet included the type, breed, sex, age and their owner
- Information of the pet owner included their name, address, home telephone number and mobile telephone number
- Information included the pet the appointment is for, the owner of the pet, the particular branches of Petcare, the attending veterinary Veterinarian and the date and time of the appointment.

- Information of the veterinary Veterinarian included their name, address, home telephone number, mobile telephone number, which branch they report to and their specialize.
- The outcome of the appointment which included the prescription and the charge made.

1. 1. 2 Entities relationship

Next step is to define the relationship of each entity. The relationship included Participation Constraint and Cardinality Ratio.

Participation Constraint

There are two types of participation constraint named ' Total' and ' Partial' participation.

A ' Total' participation (mandatory participation) constraint specifies that every member in the entity must participate as a member of another linked entity in the generalization.

A ' Partial' participation (optional participation) constraint specifies that a member of the entity does not belong to any member of another linked entity in the generalization.

Cardinality Ratio

Three types of cardinality is described as below:

One-to-one (1: 1) – Relation between two entities which is linked each other with their primary key(s) only.

Many-to-many (M: N) – Relation between two entities which is linked each other without any primary key. It also have the chance to create the redundancy error.

Therefore it is better to break into two relations such as one-to-many and many-to-one.

One-to-many (1: M) or Many-to-one (M: 1) – Relation between two entities which is linked each other from the primary key(s) to foreign key(s).

The relationship of Pet – The pet was raised by the owner; each pet should only have 1 owner but the owner can keep more then 1 pet. The owner also can keep difference kind of pets. There fore, the relationship of Pet and Owner is Total and one to many.

Relationship of Owner – The pet owner would make appointment with the Veterinarian. In the real case, an owner would make more then 1 appointment, also the Veterinarian also not handle 1 case only. A relational database is not allowed this condition. To solve the issue another entity ‘Appointment’ was added between Owner and Veterinarian. The Owner may make the appointment before visit the veterinarian, also they also can walk-in but without appointment. In order to simple the database design, we see the walk-in also as a new appointment. Therefore, the relationship of Owner and Appointment is Total and one to many.

Relationship of Appointment – the appointment was created by the pet owner and was handled by the Veterinarian. However, some Veterinarian may respond to the special case and would not handle the appointment. Therefore, the relationship of Appointment and Veterinarian is Partial and many to one relationship.

Relationship of Veterinarian – On the above we already mentioned a Veterinarian would attend more than one appointment. Also the Veterinarian would issue more than 1 diagnosis. Diagnosis is the outcome of the appointment which was issued by the veterinarian. The relationship of Veterinarian and Diagnosis is Total and one to many.

1. 1. 3 Entity Relationship Diagram (ERD)

According to the above entities and relationship design we got the first ERD as below; The ERD showed the logical database design of the Petcare. The ERD identifies a list of data elements that must be included in the database.

Fig. 1. The ERD of Petcare.

Task 2

Produce the resulting normalized tables clearly indicating the primary and foreign keys.

Answer: Normalization

After completing the logical design of the database and drawing the ERD, we can start the physical design of the database. The first thing is to create the table

and define the data-set need to store in the database. A good database design should make the content of each table is only related to its own entity. In order to meet this requirement, normalization process should be implemented.

The normalization process needs step by step to transform the database and each step have specify name, called;

Un-normal Form – UNF

First Normal Form – 1NF

Second Normal Form – 2NF

Third Normal Form – 3NF

Boyce-Codd Normal Form – BCNF

The below steps should be carried out for the normalization.

Collect the data-set

Transform the un-normalized form tables into first normal form

Transform the first normal form tables to second normal form

Transform the second normal form tables to third normal form

First Normal Form (1NF)

According the ERD design and the requirement of Petcare, the Tables of the database were created as below; also some attributes were added to

<https://assignbuster.com/entity-relationship-data-model-and-normalization-computer-science/>

identify the data-sets, e. g. PetID, which attributes also the Primary of the table.

Fig. 2. The Un-normal Form table

The above tables were included all requested data and information requested by Petcare. However, some of attributes are the same in difference tables. The tables are a un-normal form table. We need to make the content of each table is only related to its own entity. Therefore the normalization process should be implemented.

The first step is transforming the un-normalized form table into first normal form. The definition of first normal form is ' all non-key attributes in relation is if, and only if, functionally dependent upon primary key.

According the definition of first normal form, we should avulse the non-key attributes which are not relate to the primary key and create another table for this attribute.

For example; in the table Veterinarian, the Branch and Specialise are not related to the Primary key, so we need avulse these attributes and create another table, Branch & Specialise. Another table Diagnosis also the same issue.

Fig. 3. Normalized table

Fig. 4 Normalized table 2

After the process of first normalized, we got the First Normal Form table as below;

Fig. 5. First Normal Form table

Second Normal Form (2NF)

The second step is transfer first normal form to second normal form. The definition of second normal form is ' all relations are, and only if, it is in first normal form and every non-key attribute are fully functionally dependent on the primary key.

After second normalization, we got the below tables;

Fig. 6. Second Normalized tables

Third Normal Form (3NF)

The definition of third normal form is ' all relations are if, and only if, it is in second normal form and every non-key attributes in nor-transitively dependent on the primary key. Because the second normalized tables already meet the equipment. Therefore, it is third normal form.

Primary Key and Foreign Key

The next step to design the physical database is defining the Primary Key and the Foreign key. This step also defining the relationship of each table. The Primary Key is used to identify the record in each table. The Foreign Key is the key attribute to link other table.

The final normalized table with details relationship, Primary key and Foreign key were showed on the below diagram.

Fig. 7. Final table design

Data Dictionary

The last step of design the database is define the property of each attribute. The property of attribute is design the coding of the attribute. A good coding design is very important because a good coding design can let use easy to understand and take less mistake when input the data, and also determined the usability and extensibility of the database.

A good coding design should meet below requirements;

Uniqueness – only one value of the code may be applied to an entity.

Expandability – must have sufficient space for the entry of new items.

Conciseness – should use the minimum number of characters to define each item.

Uniform size & format – no addition of prefixes or suffixes.

Simplicity – simple to apply and understand.

Versatility – easy to modify for changes in conditions.

Sortability – may not be the code itself as long as it can be correlated with a sortable code.

Stability – should not need regular updating as this is costly.

Meaningful – in some circumstances it is useful for the code to reflect the characteristics of the coded entities.

Operability – the code should be sufficient to meet the current and future needs of data identification but also minimise the clerical and system resources required to operate it.

Other than the coding design, the sequence of table create also very important as the table may has relation with other tables. We should first create the table which has not relate to other tables, then create the table relate to this table. The sequence of table create list as the sequence of the Data dictionary.

According the above requirements, the tables and attributes were created and the details were listed on the Data Dictionary below;

Data Dictionary of PetcardTable NameField

NameTypeSizeStatusDescriptionOwner

OwnerID

char

6

PK

The ID of owner

a^^

O_Name

char

30

Not NULL

Name of owner

a^^

O_Address

varchar

100

NULL

Address of owner

a^^

O_Phone

char

15

NULL

Phone number of owner

a^^

O_Mobile

char

15

NULL

Mobile number of owner

a^^a^^a^^a^^a^^a^^PetType

TypeID

char

1

PK

The ID of pet type

a^^

P_Type

char

15

Not NULL

name of pet type

a^^a^^a^^a^^a^^Breed

BreedID

char

2

PK

The ID of breed type

a^^

B_type

char

15

Not NULL

Name of Breed type

a^^

TypeID

char

1

FK

relate to PetType

a^^a^^a^^a^^a^^a^^Pet

PetID

char

6

PK

The ID of the Pet

a^^

OwnerID

char

6

FK

relate to Owner

a^^

PetName

char

20

Not NULL

Name of the pet

a^^

Sex

char

1

Not NULL

sex of the pet

a^^

Age

numeric

2

Not NULL

age of the pet

a^^

Breedid

char

2

FK

relate to Breed

a^^a^^a^^a^^a^^Branch

BranchID

char

1

PK

The ID of branch

a^^

B_Name

char

15

Not NULL

Name of branch

a^^

B_Address

char

50

Not NULL

Address of branch

a^^

B_Phone

char

15

Not NULL

Phone number of branch

a^^a^^a^^a^^a^^Specialisation

SpelID

char

2

PK

The ID of specialize

a^^

Specialise

char

10

Not NULL

Name of the specialization

a^^a^^a^^a^^a^^Veterinarian

VetID

char

3

PK

The ID of veterinaria

a^^

V_Name

char

30

Not NULL

Name of veterinaria

a^^

V_Address

varchar

100

Not NULL

Address of veterinaria

a^^

V-Phone

char

15

Not NULL

Phone number of veterinaria

a^^

V_Mobile

char

15

Not NULL

Mobile number of veterinaria

a^^

SpecID

char

2

FK

relate to Specialisation

a^^

BranchID

char

1

Not NULL

The branch of the veterinaria is working

a^^a^^a^^a^^a^^Appointment

AppID

char

10

PK

The ID of appointment

a^^

VetID

char

3

FK

relate to Veterinarian

a^^

OwnerID

char

6

FK

relate to owner

a^^

BranchID

char

1

FK

relate to Branch

a^^

PetID

char

6

FK

relate to Pet

a^^

Date

date

8

Not NULL

date of appointment

a^^

Time

time

4

Not NULL

time of appointment

a^^a^^a^^a^^a^^a^^Drug Type

D_TypeID

char

2

PK

The ID of Drug Type

a^^

D_Type

char

20

Not NULL

Name of drug type

a^^a^^a^^a^^a^^Drug

DrugID

char

4

PK

The ID of drug

a^^

D_typeID

char

2

FK

relate to DrugType

a^^

DrugName

char

30

Not NULL

Name of the drug

a^^

Category

char

20

Not NULL

How to use the drug

a^^

Period

char

2

Not NULL

Number of day to use the drug

a^^a^^a^^a^^a^^Prescription

PrescID

char

10

PK

The ID of Prescription

a^^

Drug1

char

4

FK

relate to Drug

a^^

Drug2

char

4

FK

relate to Drug

a^^

Drug3

char

4

FK

relate to Drug

a^^a^^a^^a^^a^^Diagnosis

DiagID

char

6

PK

The ID of diagnosis

a^^

VelID

char

3

FK

relate to Velerinarian

a^^

PresclD

char

10

FK

relate to Prescription

Price

Numeric

6

Not NULL

The price of diagnosis

Task 3

Using a Database Management System (DBMS) set-up all of the above normalized tables, and populate them with well-designed test data..

Answer: Create tables on DBMS

After completely designed the database, we should start to create it.

Concerning to popularity and easy to use, we determined to use Microsoft Access as the Database Management System (DBMS) of the Petcare database. We should use SQL command to create the tables, attributes and relationships of tables.

<https://assignbuster.com/entity-relationship-data-model-and-normalization-computer-science/>

Create tables

The SQL command of create tables was lists below;

Create the table Owner

```
CREATE TABLE Owner
```

```
(OwnerID CHAR(6) PRIMARY KEY,
```

```
O_Name CHAR(30),
```

```
O_Address VARCHAR(100),
```

```
O_Phone CHAR(15),
```

```
O_Mobile CHAR(15) );
```

Create the table PetType

```
CREATE TABLE PetType
```

```
(TypeID CHAR(1) PRIMARY KEY,
```

```
P_Type CHAR(15) );
```

Create the table Breed

```
CREATE TABLE Breed
```

```
(BreedID CHAR(2) PRIMARY KEY,
```

```
B_type CHAR(15),
```

```
TypeID CHAR(1) REFERENCES PetType(TypeID) ON DELETE CASCADE ON  
UPDATE CASCADE );
```

Create the table PetType

```
CREATE TABLE Pet
```

```
(PetID CHAR(6) PRIMARY KEY,
```

```
OwnerID CHAR(6) REFERENCES Owner(OwnerID) ON DELETE CASCADE ON  
UPDATE CASCADE,
```

```
PetName CHAR(20),
```

```
Sex CHAR(1),
```

```
Age NUMERIC(2)
```

```
BreedID CHAR(2) REFERENCES Breed(BreedID) ON DELETE CASCADE ON  
UPDATE CASCADE );
```

All the tables also used the similar SQL command to create, we would not repeat all there. After all tables, attributes and relationships were created. The whole table diagram was showed as below figure;.

Fig. 8. Tables in MS Access

Test Data

After the table creation finished. We should input the test data to the database to test the database functioned as expended. We would input at least 5 records to each table.

The below screen showed the test data of all tables;

Fig. 9. Table Owner, Pet, PetType, Breed

Fig. 10. Table Veterinarian, Specialization, Appointment, Branch

Fig. 11. Table Prescription, Diagnosis, Drug, DrugType

The Printout of tables please refers to appendix. Task 4

Set-up and test all of the following queries using Structured Query Language (SQL).

Answer: Set-up test queries

In order to test the database design and data input correct, we need to query the data form the database. The below 2 case can test the database meet the Petcare's requirements.

Query 1Requirement:

Display the names and addresses of the branches of Petcare and the names of all the veterinary doctors working at each of the branches. Any specialism(s) of the veterinary doctors should also be shown.

This is a very general query in database, to goal the requirement we need to the join the Branch, Veterinarian and Specialization 3 tables and select the requested data. The query statement and result were listed below;

```
SELECT Branch. b_name AS Branch, Branch. b_address AS Address,  
Veterinarian. v_name AS VetName, Specialization. specialise AS  
Specialization FROM (Branch INNER JOIN Veterinarian ON Branch. branchid=  
Veterinarian. branchid) INNER JOIN Specialization ON Veterinarian. SpecID=  
Specialization. SpecID ORDER BY Branch. b_name;
```

Fig. 12. Result of Query 1.

Query 2 Requirement:

Display all the appointments for the whole of the Petcare organization. This should be ordered by date. The result should display the branch the appointment is at, the name of the veterinary doctor the appointment is with, the date and time of the appointment, the name of the animal the appointment is for, the type of animal and the breed of the animal.

Case two is more difficulty; we need to join Appointment, Branch, Veterinary, Pet, PetType and Breed total 6 tables. The query is very complex and easy to make mistake. The query statement and query results were showed below;

```
SELECT Branch. b_name AS Branch, Veterinarian. v_name AS VetName,  
Appointment. date, Appointment. time, Pet. petname AS PetName, PetType.  
P_Type AS Type, Breed. breed AS Breed FROM (((Appointment INNER JOIN  
Veterinarian ON Appointment. vetid= Veterinarian. vetid) INNER JOIN Branch  
ON Veterinarian. branchid= Branch. branchid) INNER JOIN Pet ON  
https://assignbuster.com/entity-relationship-data-model-and-normalization-computer-science/
```



```
Appointment. petid= Pet. petid)INNER JOIN Breed ON Pet. breedid= Breed.  
breedid)INNER JOIN PetType ON Breed. TypeID= PetType. TypeID)ORDER BY  
Appointment. date;
```

Fig. 13. Result of query 2

Task 5

Explain any assumptions you have made when analyzing, designing and implementing the above database..

Answer: Assumption and alternative

Each appointment ID is for one pet only. Same owner bring more than 1 pets to Petcare whould see as a new appointment.

One prescription maximum list 3 drugs. Normally 3 drugs are enough for 1 case. If the pet has more than 1 issue need more then 3 drugs, veterinary should open multi prescription for difference type of drugs. E. g. need skin medication & Painkiller.

Walk-in also sees as a new appointment.

Work Improvement

In this assignment, the database system for PatCare has not included the user interface. The database system is used for store data and the DBMS is used for manager the database or it can be sees as a bridge of Database and User interface. DBMS is not designed for user, it mainly use command line to manager the database and query the data. Even some new DBMS; system

<https://assignbuster.com/entity-relationship-data-model-and-normalization-computer-science/>

such as MS Access and SQL have GUI but also hard for a normal user to use the database and query data.

In order to let users easy to use the database, User Interface is a must. The user interface let user easy to input data, update records and check the records. And also we can per-set some forms and reports for user print out.

The user interface not need very gorgeous or complex. A good user interface is simple, convenient and easy to understand and easy to use. We should design a web base interface because the web base interface is simple and easy to manage. The most important is the web base interface does not need install additional programs, therefore not additional budget for buy other applications.