

Database design and development for petcare veterinary computer science



**ASSIGN
BUSTER**

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

Database Development Process

The process is divided into four main stages: requirements elicitation, conceptual modeling, logical modeling and physical modelling. The techniques used in the development process naturally divide into three categories: those concerned modelling, those concerned with logical modelling and those concerned with physical modelling.

Figure 1 the database development process

Requirements Elicitation

Requirements elicitation involves establishing the key technical requirements for a database system usually through formal and informal interaction between developers and organizational stakeholders such as users. It provides the structure of data needed and the use of the data in some information system context.

Stakeholder identification and participation

One of the first things that must be done in any information systems project is to identify the relevant stakeholders. A stakeholders group is any social group within and without the organization that potentially may influence the successful use and impact of the database system.

Veterinary doctor – Use the database to for managing treatment information of pet

Staff – keep track of animal and appointment

Regulators – National and regional government may need to audit the database

Requirements Elicitation and requirements specification

Requirements elicitation is the precursor to requirements specification. In terms of a system to manage Petcare, the following is a list of proposed requirements for system:

The database should be capable of supporting the following transaction:

Create and maintain records recording the details of Petcare Pets clinics and the members of staff at each clinic.

Create and maintain records recording the details of pet owners.

Create and maintain the details of pets.

Create and maintain records recording the details of the types of treatments for pets.

Create and maintain records recording the details of examinations and treatments given to pets.

Create and maintain records recording the details of invoices to pet owners for treatment to their pets.

Create and maintain pet owner/pet appointments at each clinic.

Data requirements PetCare veterinary surgery

Petcare has six medium sized veterinary surgery clinics across London. The details of each clinic include address of branch, telephone number, opening hours and emergency contact telephone number. Assuming that each clinic has a number of staff for example vets, nurses, secretaries and cleaners.

Staff

The details stored on each member of staff include the staff name, address, home telephone number and mobile telephone number.

Pet owners

When a pet owner first contacts a clinic of Petcare the details of the pet owner are recorded, which include name, address, home telephone number and mobile telephone number.

Pets

The details of the pet requiring treatment are noted, which include a put number, type of pet, age and sex.

Examinations

When a sick pet is brought to a clinic, the vet on duty examines the pet. The details of each examination are recorded and include an examination number, the date and time of the examination, the name of the vet, the pet

number, pet name, and type of pet, and a full description of the outcome of the examination results.

Petcare provides various treatments for all types of pets. The details of each treatment include a treatment number, full description of the treatment, and the cost to the pet owner. Based on the result of the examination of a sick pet, the vet may propose one or more types of treatment. For each types of treatment, the information recorded includes the examination number and date.

Invoices

The pet owner is responsible for the cost of the treatment given to a pet. The owner is invoiced for the treatment arising from each examination, and the details recorded on the invoice include the invoice number, invoice date, owner number, owner name and full address, put number, put name, and details of the treatment given. The invoice provides the cost for each type of treatment and total cost of all treatments given to the pet. Additional data is also recorded on the payment for example cash, credit card or check.

Appointments

If the pet requires to be seen by the vet at later date, the owner and pet are given an appointment. The details of an appointment are recorded and include an appointment number, owner name, date and time.

Using the logical database design methodology

Entity-Relationship modeling is a top-down approach to database design. We begin ER modeling by identifying the important data (called entities) and relationships between the data that must be represented in the model. We then add more details such as the information we want to hold about the entities and relationships (called attributes) and any constraints on the entities, relationships, and attributes. Identify entities

Identify entities

A set of objects with the same properties, which are identified by a user or organization as having an independent existence

The first step in logical database design is to identify entity that you have to represent in the database.

Entity name

Description

Occurrence

Clinic

Veterinary clinics

One or more Petcare clinics located throughout London

Staff

General term describing all staff employed by Petcare

Each member of staff works at a particular clinic

PetOwner

Owners of pets taken to Petcare

Owner takes his/her pet to a particular clinic

Pet

Sick animal seek treatment to the clinic

One or more animal are taken to the clinic

Examination

Treatment

Invoice

Appointment

PetTreatment

Attributes

The particular properties of entities are called attributes. Attributes represent what we want to know about entities.

Single-Valued attributes

The majority of attributes are single-valued for a particular entity. For example, each occurrence of the Staff entity has a single value for the

<https://assignbuster.com/database-design-and-development-for-petcare-veterinary-computer-science/>

staffNo attribute (for example, 001), and therefore the staffNo attribute is referred to as being single-valued.

Derived attributes

An attribute that represents a value that is derivable from the value of a related attribute, or set of attributes, not necessarily in the same entity.

Some attributes may be related for a particular entity. For example, the age of a pet is derivable from the date of birth (DOB) attribute, and therefore the age and DOB attributes are related. We refer to the age attribute as a derived attribute, the value of which is derived from the DOB attribute.

Age is not normally stored in a database because it would have to be updated regularly. On the other hand, as the date of birth never changes and age can be derived from the date of birth, the date of birth is stored instead, and age is derived from the DOB attribute, when needed.

Identify relationships

Having identified the entities; the next step is to identify all the relationships that exist between these entities. A relationship is a set of associations between participating entities. As with entities, each association should be uniquely identifiable within the set. A uniquely identifiable association is called a relationship occurrence.

Clinic Relationships Entity Relationship Entity

Clinic

Has

Veterinary

Registers

Pet

Schedules

Appointment

IsContactedBy

PetOwner

Veterinary RelationshipsEntityRelationshipEntity

Veterinary

Performs

Examination

Pet Owner RelationshipsEntityRelationshipEntity

PetOwner

Owns

Pet

Pays

Invoice

Attends

Appointment

Pet RelationshipsEntityRelationshipEntity

Pet

Undergoes

Examination

Attends

Appointment

Cardinality Ratio of PetCare database

Cardinality or degree concerns the number of instances involved in a relationship. A relationship can be said to be either a 1: 1 (one-to-one) relationship, a 1: M (one-to-many) relationship, or an M: N (many-to-many) relationship.

Final Clinic relationshipsEntityCardinalityRelationshipCardinalityEntity

Clinic

1.. 1

Has

1.. M

Staff

1.. M

Registers

1.. M

Pet

1.. 1

Schedules

1...M

Appointment

1.. 1

IsContactedBy

1.. M

PetOwner

Final Veterinary RelationshipsEntityCardinalityRelationshipCardinalityEntity

Veterinary

1.. 1

Performs

1.. M

Examination

Final Pet Owner RelationshipsEntityCardinalityRelationshipCardinalityEntity

PetOwner

1.. 1

Owns

1.. M

Pet

1.. 1

Pays

1.. M

Invoice

1.. 1

Attends

1.. M

Appointment

Final Pet Relationships Entity Cardinality Relationship Cardinality Entity

Pet

1.. 1

Undergoes

1.. M

Examination

1.. 1

Attends

1.. M

Appointment

First draft Entity Relationship Data Model
Second draft Entity Relationship
Data Model
Final Entity Relationship Data Model
Task 2 Normalization

Normalization is a technique for producing a set of table with desirable properties that support the requirements of a user or company. There are several normal forms, although the most ones are called first normal form (1NF), second normal form (2NF), and third normal form (3NF). All these normal forms are based on rules about relationships among the columns of a table.

First normal forms (1NF)

<https://assignbuster.com/database-design-and-development-for-petcare-veterinary-computer-science/>

Only first normal form (1NF) is critical in creating appropriate tables for relational databases. All the subsequent normal forms are optional. A table in which the intersection of every column and record contains only one value

Clinic (clinicNo, address, city, state, zipcode,

telNo, faxNo, openingHour)

Primary Key clinicNo

Alternate Key zipCode

Alternate Key telNo

Alternate Key faxNo

Alternate Key openingHour

clinicNoaddressstelNoopeningHour

C001

Enfield

503-555-3618, 503-555-2727, 503-555-6534

9. 00-21-00

C002

Islington

206-555-6756, 206-555-8836

9. 00-21-00

C003

Hackney

212-371-3000

9. 00-21-00

C004

Holloway

206-555-3131, 206-555-4112

9. 00-21-00

C005

Chingford

8502333

9. 00-21-00

C006

Leyton

4650000

9. 00-21-00 This version of the Clinic table is not in 1NF

More than one value, so not in 1NF

Converting to 1NF

To convert this version of the Clinic table to 1NF, we create separate table called ClinicTelephone to hold the telephone number of clinics, by removing the telNo column from the Clinic table along with a copy of the primary key of the Clinic table. The primary key for the new ClinicTelephone table is now the telNo column. The Clinic and ClinicTelephone table are in 1 NF as there is a single value at the intersection of every column with every record for each table

Clinic (Not 1NF) clinicNo address telNo openingHour

C001

Enfield

503-555-3618, 503-555-2727, 503-555-6534

9. 00-21-00

C002

Islington

206-555-6756, 206-555-8836

9. 00-21-00

C003

Hackney

212-371-3000

9. 00-21-00

C004

Holloway

206-555-3131, 206-555-4112

9. 00-21-00

C005

Chingford

8502333

9. 00-21-00

C006

Leyton

4650000

9. 00-21-00

Remove telNo column and create a new column called telNo in the new table

Take copy of clinicNo column to new table to become foreign key

ClinicTelephone (1NF)clinicNoteIDNo

C001

503-555-3618

C001

503-555-2727

C001

503-555-6534

C002

206-555-6756

C002

206-555-8836

C003

212-371-3000

C004

206-555-3131

C004

206-555-4112

C005

8502333

C006

4650000

Clinic (1NF)clinicNoaddressopeningHour

C001

Enfield

9. 00-21-00

C002

Islington

9. 00-21-00

C003

Hackney

9. 00-21-00

C004

Holloway

9. 00-21-00

C005

Chingford

9. 00-21-00

C006

Leyton

9. 00-21-00

Second normal form (2NF)

Second normal form applies only to tables with composite primary keys that are table with a primary key composed of two or more columns. A 1NF table with a single column primary key is automatically in at least 2NF. A table that is not in 2NF may suffer from update anomalies. A table that is already in 1NF and which the values in each non-primary-key column can be worked out from values in all columns that make up the primary key.

Third normal form (3NF)

Although 2NF table have less redundancy that table in 1 NF, they may still suffer from update anomalies. A table that is already in 1NF and 2NF, and in which the values in all non-primary key columns can be worked out from only the primary key column and no other columns.

staffNo

name

<https://assignbuster.com/database-design-and-development-for-petcare-veterinary-computer-science/>

lastName

salary

clinicNo

clinicAddress

telNo

S001

Tom

Adams

25000

C001

Enfield

5035553618

S002

Sally

Daniels

35222

C001

Enfield

5035553618

S003

Mary

Chin

5200

C002

Islington

206555675

S004

Sally

Stern

5000

C002

Islington

206555

S005

Art

Peters

45822

C003

Hackney

8502333

S006

Tommy

Verciti

65000

C004

Holloway

4650000

Values in clinicNo and clinicAddress columns can be worked out from telNo,
so table not in 3NF

Values in clinicNo and telNo columns can be worked out from clinicAddress,
so table not in 3NF

Values in all non-primary-key columns can be worked out from the primary key, staffNo

Values in clinicAddress and telNo columns can be worked out from clinicNo, so table not in 3NF

staffNo

name

lastName

salary

clinicNo

clinicAddress

telNo

StaffClinic (Not 3NF)

staffNo

name

lastName

salary

clinicNo

clinicAddress

telNo

S001

Tom

Adams

25000

C001

Enfield

5035553618

S002

Sally

Daniels

35222

C001

Enfield

5035553618

S003

Mary

Chin

5200

C002

Islington

206555675

S004

Sally

Stern

5000

C002

Islington

206555

S005

Art

Peters

45822

C003

Hackney

8502333

S006

Tommy

Verciti

65000

C004

Holloway

4650000

Move column to new table

Take copy of clinicNo column to new table to become primary key

Clinic (3NF)

clinicNo

clinicAddress

telNo

C001

Enfield

5035553618

C001

Enfield

5035553618

C002

Islington

206555675

C002

Islington

206555

C003

Hackney

8502333

C004

Holloway

4650000

Staff (3NF)

staffNo

name

lastName

salary

clinicNo

S001

Tom

Adams

25000

C001

S002

Sally

Daniels

35222

C001

S003

Mary

Chin

5200

C002

S004

Sally

Stern

5000

C002

S005

Art

Peters

45822

C003

S006

Tommy

Verciti

65000

C004

Primary key

Becomes foreign key

Becomes candidate key

Becomes primary key

Task 3

Using a Database Management System (DBMS) of your choice, set up all the above normalized tables, and populate them with well-designed test data (minimum 5 records per table). Provides printouts of all tables.

Reasonable assumption may be made with regard to data

Clinic Table

```
create table clinic
```

```
(
```

```
clinicNO int not null primary key,
```

```
telNo varchar(255),
```

```
address varchar(255),
```

```
)
```

```
alter table clinic
```

<https://assignbuster.com/database-design-and-development-for-petcare-veterinary-computer-science/>

```
add clinicName varchar (255)
```

```
alter table clinic
```

```
add openingHour varchar(255)
```

```
alter table clinic
```

```
add eTelNo varchar (255)
```

```
insert into clinic
```

```
values (01, 2863015,'Darwin Avenue','8. 00-21. 00?', 2863000,'Enfield')
```

```
insert into clinic
```

```
values (02, 4650001,'John David Avenue','8. 00-21. 00?', 2868000,'Islington')
```

```
insert into clinic
```

```
values (03, 4278926,'King Arthur Avenue','8. 00-21. 00?',
```

```
2867000,'Hackney')
```

```
insert into clinic
```

```
values (04, 2682365,'Paul Mac Avenue','8. 00-21. 00?', 2866000,'Holloway')
```

```
insert into clinic
```

```
values (05, 4682685,'James Micheal Avenue','8. 00-21. 00?',
```

```
2865000,'Chingford')
```

```
insert into clinic
```

<https://assignbuster.com/database-design-and-development-for-petcare-veterinary-computer-science/>

values (06, 2863015,'Benoit Frank Avenue', '8. 00-21. 00?',
2864000,'Leyton')

Pet owner

create table petowner

(

ownerID int not null primary key,

oFName varchar (255),

oLName varchar (255),

clinicNo int foreign key references clinic (clinicNo))

alter table petowner

add address varchar(255)

alter table petowner

add hTelNo varchar(255)

alter table petowner

add mTelNo varchar (255)

insert into petowner

values (01,'Marvin', 'Hemraj', 1,'Edith Cavel Str', 2106584, 758956)

```
insert into petowner
```

```
values (02,'Ramjeet','Lavin', 2,'Avenue Gonin', 2564589, 7585695)
```

```
insert into petowner
```

```
values (03,'Arzeena','Bakarkhan', 3,'Gorgetown Str', 2106584, 758956)
```

```
insert into petowner
```

```
values (04,'Chetan','Sing', 4,'Jackson Road', 2458695, 7582658)
```

```
insert into petowner
```

```
values (05,'Hansley','Nowjee', 5,'15 Ollier Avenue', 2565458, 7589562)
```

```
insert into petowner
```

```
values (06,'Sam','Fisher', 6,'Leess Street', 26584585, 75895623)
```

Pet Table

```
create table pet
```

```
(
```

```
petNo int not null primary key,
```

```
type varchar (255),
```

```
breed varchar (255),
```

```
sex varchar (255),
```

```
dob varchar (255)
```

```
)
```

```
alter table pet
```

```
add clinicNo int foreign key references clinic (clinicNo)
```

```
alter table pet
```

```
add ownerid int foreign key references petOwner (ownerid)
```

```
alter table pet
```

```
add petName varchar (255)
```

```
insert into pet
```

```
values (01,'Dog','Terroer','Male','1 Jan 2004?', 01, 01,'Wouf')
```

```
insert into pet
```

```
values (02,'Dog','Poodle','Female','2 Feb 2005?', 02, 02,'Snoopy')
```

```
insert into pet
```

```
values (03,'Cat','Persian','Male','3 March 2006?', 03, 03,'Minous')
```

```
insert into pet
```

```
values (04,'Cat','Siamese','Female','4 April 2007?', 04, 04,'Milous')
```

```
insert into pet
```

```
values (05,'Rabbit','Dwarf','male','5 May 2008?', 05, 05,'Lapino')
```

```
insert into pet
```

```
values (06,'Cat','Siamese','Female','4 June 2009?', 06, 06,'Lapinas')
```

Examination Table

```
create table examination
```

```
(
```

```
examNo int not null primary key )
```

```
alter table examination
```

```
add veterinaryId int foreign key references veterinary (veterinaryId)
```

```
alter table examination
```

```
add petNo int foreign key references pet (petNo)
```

```
alter table examination
```

```
add presDrugType varchar (255),
```

```
presPeriod varchar (255)
```

```
insert into examination
```

```
values (1, 1,'Anti-biotic', 15, 01)
```

```
insert into examination
```

```
values (2, 2,'Painkiller', 14, 02)
```

```
insert into examination
```

```
values (3, 3,'Behaviour modification', 13, 03)
```

```
insert into examination
```

```
values (4, 4,'Ear medication', 12, 04)
```

```
insert into examination
```

```
values (5, 5,'Skin medication', 15, 05)
```

```
insert into examination
```

```
values (6, 6,'Painkiller', 10, 06)
```

```
Appointment Table
```

```
create table appointment
```

```
(
```

```
appNo int not null primary key,
```

```
aDate varchar (255),
```

```
aTime varchar (255),
```

```
petNo int foreign key references pet (petNo),
```

```
ownerID int foreign key references petowner (ownerID),
```

```
veterinaryId int foreign key references veterinary (veterinaryId)
```

```
)
```

```
alter table appointment
```

```
add clinicNO int foreign key references clinic (clinicNO)
```

```
alter table appointment
```

```
alter column aDate date
```

```
insert into appointment
```

```
values (01,'2 October 2010?','10. 00?', 1, 1, 1, 1)
```

```
insert into appointment
```

```
values (02,'3 November 2009?','10. 35?', 2, 2, 2, 2)
```

```
insert into appointment
```

```
values (03,'4 December 2009?','13. 00?', 3, 3, 3, 3)
```

```
insert into appointment
```

```
values (04,'5 January 2010?','15. 00?', 4, 4, 4, 4)
```

```
insert into appointment
```

```
values (05,'6 Feb 2010?','18. 00?', 5, 5, 5, 5)
```

```
insert into appointment
```

values (06,'7 March 2010?,'9. 00?, 6, 6, 6, 6)

Invoice Table

```
create table invoice
```

```
(
```

```
invoiceNo int not null primary key,
```

```
ownerid int foreign key references petowner (ownerid),
```

```
amt int
```

```
)
```

```
alter table invoice
```

```
add examNo int foreign key references examination (examNo)
```

```
insert into invoice
```

```
values (01, 01, 500, 1)
```

```
insert into invoice
```

```
values (02, 02, 2000, 2)
```

```
insert into invoice
```

```
values (03, 03, 400, 3)
```

```
insert into invoice
```

```
values (04, 04, 300, 4)
```

```
insert into invoice
```

```
values (05, 05, 1500, 5)
```

```
insert into invoice
```

```
values (06, 06, 750, 6)
```

Veterinary Table

```
create table veterinary
```

```
(
```

```
veterinaryId int not null primary key,
```

```
name varchar (255),
```

```
Address varchar(255),
```

```
hTelNo varchar(255),
```

```
mTelNo varchar(255),
```

```
clinicBranch varchar (255),
```

```
specialise varchar(255)
```

```
)
```

```
select * from veterinary
```



```
alter table veterinary
```

```
add spspecialise varchar(255)
```

```
alter table veterinary
```

```
drop column clinicBranch
```

```
alter table veterinary
```

```
add clinicNo int foreign key references clinic (clinicNo)
```

```
insert into veterinary
```

```
values (01,'Jean','High Street Way 43?', 6358264, 7595865, 1,'dog')
```

```
insert into veterinary
```

```
values (02,'Robbin','Lower Downtown 2?', 6582354, 7362548, 2,'cat')
```

```
insert into veterinary
```

```
values (03,'Ricky','Market Ville Road', 4582356, 7586523, 3,'rabit')
```

```
insert into veterinary
```

```
values (04,'Rowan','Little China Road2?', 4582653, 7263158, 4,'dog')
```

```
insert into veterinary
```

```
values (05,'Laksh','Havana Roadway', 8596564, 7236458, 5,'rabit')
```

```
insert into veterinary
```

values (06,'Hans','Talipos Road', 4625687, 7859584, 6,'cat')

Task 4

Set-up and test all of the following queries using Structured Query Language (SQL). Provide printouts of SQL code for each query and the output produced when you run the query in the database you have developed.

Query Question 1

Display the names and address of the branches of Petcare and the name of all veterinary doctors working at each of the branches. Any specialism (s) of the veterinary doctors should also show.

Using SQL Code

```
SELECT clinic. clinicName AS [Branches Name], clinic. address AS [Branches  
Address], veterinary. name AS [Veterinary Doctor Name],  
veterinary. specialise AS [Veterinary Specialism]
```

```
FROM clinic INNER JOIN
```

```
veterinary ON clinic. clinicNO = veterinary. clinicNo
```

The Result

Using Query Designer

The Result

Query Question 2

<https://assignbuster.com/database-design-and-development-for-petcare-veterinary-computer-science/>

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.

Using SQL code

```
SELECT clinic. clinicName AS [Branches Name], veterinary. name AS  
[Veterinary Doctor Name], appointment. aDate AS [Appointment Date],  
appointment. aTime AS [Appointment Time], pet. petName AS [Pet Name],  
pet. type AS [Pet Type], pet. breed AS [Pet Breed]  
  
FROM appointment INNER JOIN  
  
clinic ON appointment. clinicNO = clinic. clinicNO INNER JOIN  
  
veterinary ON appointment. veterinaryId = veterinary. veterinaryId AND  
clinic. clinicNO = veterinary. clinicNo INNER JOIN  
  
pet ON appointment. petNo = pet. petNo AND clinic. clinicNO = pet. clinicNo  
  
ORDER BY [Appointment Date]
```

The Result

Using Query Designer

The Result

Task 5

Explain any assumptions you have made when analyzing, designing and implementing the above database, justify the approach you have taken and explain any alternative approaches you could have taken to any of the above tasks. Discuss any changes you would make to improve your work.

Determine candidate, primary and alternate key attributes

This step is concerned with identifying the candidate key for an entity and then selecting one to be the primary key. In the process of identifying primary keys, note whether an entity is strong or weak.

In trying to identify candidate keys, I observe that the clinic number for the Clinic entity, the veterinary number for the veterinary entity, the invoice number for the Invoice entity are unique for the entire practice. On the other hand, the owner number for the PetOwner entity, the pet number for the Pet entity, are only unique for a particular clinic. It's not uncommon for a company to give different offices a degree of local autonomy. However, in a centralized database system it's sometimes more appropriate to have uniqueness throughout the company. In discussion with the PetCare management, it's agreed that all numbers should be allocated across the entire practice, as opposed to each branches. If this had not been the decision, it would have been necessary to add the clinic number to those numbers only unique within each clinic to gain uniqueness across the practice. With this in mind, I have now identify the primary keys and foreign key.

Check model for redundancy

At this point, I have a logical data model for Petcare. However, the data model may contain some redundancy which should be removed. More specifically, I have to:

- (1) Re-examine one-to-one (1: 1) relationships.
- (2) Remove redundant relationships.
- (3) One-to-one (1: 1) relationships

Redundant relationships

There are a number of relationships between PetOwner, Pet, Clinic, and Appointment, and a closer examination is useful to identify any redundant relationships. First of all, note that the PetOwner/Pet entities have mandatory participation in the POAttends/PAttends/Owns relationships, and that a PetOwner may own many pets. Therefore, for any given Appointment we can identify the Owner through the POAttends relationship, but we cannot then identify the Pet through the Owns relationship. However, for any given Appointment, we can identify the Pet through the PAttends relationship and for any given Pet we can identify the PetOwner through the Owns relationship, which suggests that the POAttends relationship is redundant. In a similar way, through the PAttends relationship we can identify the Pet, and through the Registers relationship we can identify the Clinic involved in the Appointment, which suggests the Schedules relationship is also redundant.

Note that the IsContactedBy relationship between Clinic and PetOwner also appears to be redundant. However, PetCare notes the details of pet owners when they first make contact and only obtains the details of pets at the first appointment, and so the IsContactedBy relationship is retained.

Check business rules

Business rules are the constraints that I have impose in order to protect the database from becoming inconsistent. Of the six types of business rules, four were identified in previous steps and documented in the above. I consider the remaining two here: referential integrity and other business rules.

Referential integrity

There are two issues to consider here:

(1) Identify whether nulls are allowed for the foreign key. In general, if the participation of the child table in the relationship is mandatory, then the strategy is that nulls are not allowed. On the other hand, if the participation of the child table is optional, then nulls should be allowed.

(2) Identify the existence constraints under which a foreign key may be inserted, updated, or deleted. In general, this involves specifying two actions for each foreign key: an ON UPDATE action and an ON DELETE action.

Other business rules

Finally, consider whether there are any other types of constraints that PetCare has defined that have not been covered elsewhere in the data model.

Such constraints are more generally called business rules.

The referential integrity constraints for PetCare table. Clinic

Foreign Key veterinary references Staff(veterinaryNo) ON UPDATE CASCADE
ON DELETE NO ACTION

Veterinary

Foreign Key clinicNo references Clinic(clinicNo) ON UPDATE CASCADE ON
DELETE NO ACTION

PetOwner

Foreign Key clinicNo references Clinic(clinicNo) ON UPDATE CASCADE ON
DELETE NO ACTION

Pet

Foreign Key ownerNo references Owner(ownerNo) ON UPDATE CASCADE ON
DELETE CASCADE

Foreign Key clinicNo references Clinic(clinicNo) ON UPDATE CASCADE ON
DELETE NO ACTION

Examination

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

Foreign Key staffNo references Staff(staffNo) ON UPDATE CASCADE ON DELETE NO ACTION

Invoice

Foreign Key ownerNo references Owner(ownerNo) ON UPDATE CASCADE ON DELETE NO ACTION

Foreign Key examNo references Examination(examNo) ON UPDATE CASCADE ON DELETE NO ACTION

Appointment

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

Review logical database design with users

The logical database design is now complete and fully documented. At this point, you should review the logical data model and supporting documentation with the users. We'll assume that this does not identify any major deficiencies in the design. This completes the logical database design methodology for Petcare.

Design base tables

During logical database design you produced a design for a number of base tables to represent the entities and relationships in the logical data model.

This included a description of:

- each table, its attributes, the primary, alternate, and foreign keys, and integrity constraints;

- each attribute its domain, an optional default value, whether it can hold nulls, and whether it's derived.

The design of the base tables also included a definition of domains, default values, and null indicators. For example, for the Pet table of Petcare, you may produce the design using an extended Database Design Language (DBDL). I use this information to determine how to implement the base tables in the target DBMS, which for this time will be Microsoft SQL Server 2008.