2.32 *Examine the data in the Veterinary Office List—Version One in Figure 1-30 (see page 52), and state assumptions about functional dependencies in that table. What is the danger of making such conclusions on the basis of sample data?*

PetName → (PetType, PetBreed, PetDOB, OwnerLastName, OwnerFirstName, OwnerPhone, OwnerEmail)

OwnerEmail → (OwnerLastName, OwnerFirstName, OwnerPhone)

OwnerPhone → (OwnerLastName, OwnerFirstName, OwnerEmail)

The danger is that there may be possibilities not apparent from sample data. For example, two owners might have pets with the same name.

2.33 *Using the assumptions you stated in your answer to question 2.32, what are the determinants of this relation? What attribute(s) can be the primary key of this relation?*

Attributes that can be the primary key are called candidate keys.

Determinants: **PetName, OwnerEmail, OwnerPhone**

Candidate keys: **PetName**

2.34 *Describe a modification problem that occurs when changing data in the relation in question 2.32 and a second modification problem that occurs when deleting data in this relation.*

Changes to owner data may need to be made in several rows.

Deleting data for the last pet of an owner deletes owner data as well.

ANSWERS TO EXERCISES

2.38 *Apply the normalization process to the Veterinary Office List—Version One relation shown in Figure 1-30 (see page 55) to develop a set of normalized relations. Show the results of each of the steps in the normalization process.*

STEP ONE:

PET-AND-OWNER (PetName, PetType, PetBreed, PetDOB, OwnerLastName, OwnerFirstName, OwnerPhone, OwnerEmail)

Functional Dependencies:

PetName → (PetType, PetBreed, PetDOB, OwnerLastName, OwnerFirstName, OwnerPhone, OwnerEmail)

OwnerEmail → (OwnerLastName, OwnerFirstName, OwnerPhone)

OwnerPhone → (OwnerLastName, OwnerFirstName, OwnerEmail)

PET-AND-OWNER Candidate Keys: **PetName**

Is every determinant a candidate key?

NO—OwnerEmail and OwnerPhone are NOT candidate keys.

<u>STEP TWO:</u>

Break into two relations: OWNER and PET

OWNER (OwnerLastName, OwnerFirstName, OwnerPhone, OwnerEmail)

PET (<u>PetName</u>, PetType, PetBreed, PetDOB, {Foreign Key})

FOR OWNER:

Functional Dependencies:

OwnerEmail → (OwnerLastName, OwnerFirstName, OwnerPhone)

OwnerPhone → (OwnerLastName, OwnerFirstName, OwnerEmail)

OWNER Candidate Keys: **OwnerPhone, OwnerEmail**

Is every determinant a candidate key?

YES—OwnerEmail and OwnerPhone are candidate keys—Normalization complete!

We can choose either candidate key as primary key.

(A) IF WE USE OwnerPhone as primary key, THEN:

OWNER (<u>OwnerPhone</u>, OwnerLastName, OwnerFirstName, OwnerEmail)

PET (<u>PetName</u>, PetType, PetBreed, PetDOB, *OwnerPhone*)

Functional Dependencies:

PetName → (PetType, PetBreed, PetDOB, OwnerPhone)

PET Candidate Keys: **PetName**

Is every determinant a candidate key?

YES—PetName is a candidate key—Normalization complete!

FINAL NORMALIZED REALTIONS:

OWNER (<u>OwnerPhone</u>, OwnerLastName, OwnerFirstName, OwnerEmail)

PET (<u>PetName</u>, PetType, PetBreed, PetDOB, *OwnerPhone*)

(B) IF WE USE OwnerEmail as primary key, THEN:

OWNER (OwnerPhone, OwnerLastName, OwnerFirstName, <u>OwnerEmail</u>)

PET (<u>PetName</u>, PetType, PetBreed, PetDOB, *OwnerEmail*)

Functional Dependencies:

PetName → (PetType, PetBreed, PetDOB, OwnerEmail)

PET Candidate Keys: **PetName**

Is every determinant a candidate key?

YES—PetName is a candidate key—Normalization complete!

FINAL NORMALIZED REALTIONS:

OWNER (OwnerPhone, OwnerLastName, OwnerFirstName, OwnerEmail)

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

2.40 *Consider the following relation:*

STUDENT (StudentNumber, StudentName, SiblingName, Major)

Assume that the values of SiblingName are the names of all of a given student's brothers and sisters; also assume that students have at most one major.

a. *Show an example of this relation for two students, one of whom has three siblings and the other of whom has only two siblings.*

StudentNumber	StudentName	SiblingName	Major
100	Mary Jones	Victoria	Accounting
100	Mary Jones	Slim	Accounting
100	Mary Jones	Reginald	Accounting
200	Fred Willows	Rex	Finance
200	Fred Willows	Billy	Finance

b. *List the candidate keys in this relation.*

STUDENT Candidate Keys: **(StudentNumber, SiblingName)**

This assumes that StudentName is not unique.

c. *State the functional dependencies in this relation.*

StudentNumber → (StudentName, Major)

(StudentNumber, SiblingName) → (StudentName, Major)

d. *Explain why this relation does not meet the relational design criteria set out in this chapter (i.e., why this is not a well-formed relation).*

Some attributes are functionally dependent on a part of the composite primary key.

e. *Divide this relation into a set of relations that meet the relational design criteria (that is, that are well formed).*

Break into two relations: STUDENT and STUDENT-SIBLING

STUDENT (StudentNumber, StudentName, Major)

STUDENT-SIBLING (*StudentNumber*, SiblingName)

FOR STUDENT-SIBLING:

Functional Dependencies:

(StudentNumber, SiblingName) → (StudentNumber)

(StudentNumber, SiblingName) → (SiblingName)

STUDENT-SIBLING Candidate Keys: **(StudentNumber, SiblingName)**

Is every determinant a candidate key?

YES—(StudentNum, SiblingName) is a candidate key—Normalization complete!

FOR STUDENT:

STUDENT (StudentNumber, StudentName, Major)

Functional Dependencies:

StudentNumber → (StudentName, Major)

STUDENT Candidate Keys: **StudentNumber**

Is every determinant a candidate key?

YES—StudentNumber is a candidate key—Normalization complete!

FINAL NORMALIZED REALTIONs:

STUDENT (StudentNumber, StudentName, Major)

STUDENT-SIBLING (*StudentNumber*, SiblingName)

2.43 Consider a table named ORDER_ITEM, with data as shown in Figure 2-26. The schema for ORDER_ITEM is:

	OrderNumber	SKU	Quantity	Price
1	1000	201000	1	300.00
2	1000	202000	1	130.00
3	2000	101100	4	50.00
4	2000	101200	2	50.00
5	3000	100200	1	300.00
6	3000	101100	2	50.00
7	3000	101200	1	50.00

ORDER_ITEM (OrderNumber, SKU, Quantity, Price)

Where SKU is a "Stocking Keeping Unit" number, which is similar to a part number. Here it indicates which product was sold on each line of the table. Note that one OrderNumber must have at least one SKU associated with it, and may have several. Use this table and the detailed discussion of normal forms of pages 88-89 to answer the following questions.

a. Define 1NF. Is ORDER_ITEM in 1NF? If not, why not, and what would have to be done to put it into 1NF? Make any changes necessary to put ORDER_ITEM into 1NF. If this step requires you to create an additional table, make sure that the new table is also in 1NF.

First Normal Form is any table that meets the definition of a relation (Figure 2.1 below). ORDER_ITEM is in 1NF.

1. Rows contain data about an entity
2. Columns contain data about attributes of the entity
3. Cells of the table hold a single value
4. All entries in a column are of the same kind
5. Each column has a unique name
6. The order of the columns is unimportant
7. The order of the rows is unimportant
8. No two rows may hold identical sets of data values

b. Define 2NF. Now that ORDER_ITEM is in 1NF, is it also in 2NF? If not, why not, and what would have to be done to put it into 2NF? Make any changes necessary to put ORDER_ITEM into 2NF. If this step requires you to create an additional table, make sure that the new table is also in 2NF.

Second Normal Form is any table that 1) in First Normal Form, and 2) all nonkey attributes are determined by the entire primary key. In this case, the Primary Key is OrderNumber, SKU. Because SKU alone determines Price (SKU → Price), ORDER_ITEM is NOT in 2NF.

This is solved by creating another table (PRODUCT), where SKU is both the Primary Key and Foreign Key in PRODUCT, and Price is moved out of ORDER_ITEM and into the PRODUCT Table.

ORDER_ITEM (OrderNumber, SKU, Quantity)

PRODUCT (SKU, Price)

c. Define 3NF. Now that ORDER_ITEM is in 2NF, is it also in 3NF? If not, why not, and what would have to be done to put it into 3NF? Make any changes necessary to put ORDER_ITEM into 3NF. If this step requires you to create an additional table, make sure that the new table and any other tables created in previous steps are also in 3NF.

Third Normal Form is any table that 1) in Second Normal Form, and 2) and no nonkey attributes are determined by any other nonkey attributes. Because the original question was not in Second Normal Form, it was NOT in Third Normal Form. The solution in B fixes this problem, and then both ORDER_ITEM and PRODUCT are in Third Normal Form.

d. Define BCNF. Now that ORDER_ITEM is in 3NF, is it also in BCNF? If not, why not, and what would have to be done to put it into BCNF? Make any changes necessary to put ORDER_ITEM into BCNF. If this step requires you to create an additional table, make sure that the new table and any other tables created in previous steps are also in BCNF.

BCNF is any table that 1) in Third Normal Form, and 2) and all determinates are candidate keys. Because the original question was not in Second Normal Form, it was NOT in BCNF. The solution in B fixes this problem, and then both ORDER_ITEM and PRODUCT are in BCNF.