Chapter 2 – Conceptual Data Modeling

Chapter 2 Objectives

After completing this chapter, the student will understand:

- The fundamental terms (i.e., grammar) associated with the Entity-Relationship Model
  - Entity type
  - Attribute
  - Integrity constraints as technical expressions of business rules
  - Relationship type
  - Structural constraints of a relationship type
  - Base entity type versus weak entity type
  - Cluster entity type
  - Associative entity type
  - Deletion rules/constraints
- Common errors that occur during the development of data models

Chapter 2 Overview

This chapter introduces the fundamental constructs and rules for conceptual data modeling using the entity-relationship (ER) modeling grammar as the modeling tool. The basic units of the ER model, that is, entity type, entity class, attribute, unique identifier, and relationship type, are treated in detail. This is followed by exemplified descriptions of weak entity type, cluster entity type and associative entity type. Finally, the role of deletion rules and the specification of deletion constraints are discussed.

Chapter 2 Key Terms

Method	In conceptual modeling, describes how to use the conceptual modeling grammar.
Semantic modeling	The overall activity of attempting to represent meaning.
Conceptual modeling script	The product of conceptual "data" modeling. We just say conceptual modeling in the book.
Conceptual modeling grammar	Defines the set of constructs and rules for the conceptual modeling of a phenomenon.
Object type	A named collection of properties that sufficiently describes an actual distinct type of identity in the real world.
Object	An actual occurrence of an object type (also referred to as an object instance or object occurrence).
Entity type	The conceptual representation of an object type.
Entity	The occurrence of an entity type (also referred to as an entity instance or entity occurrence).
Object class	A generalization of different related object types that have shared properties.
Entity class	A generalization of different related entity types that have shared attributes.
Entity set	A collection of entities.
Property	A fundamental characteristic of an object type.
Attribute	A fundamental characteristic of an entity type (i.e., the conceptual representation of a property).
Property value set	The collection of all possible values of a property.
Domain	The collection of all possible values of an attribute.
Data type	A characteristic of an attribute; common data types include numeric, alphabetic, alphanumeric, logical, and date.
Simple attribute	An attribute that has a discrete factual value; a simple attribute is sometimes referred to as an atomic attribute.
Composite attribute	An attribute that can be meaningfully subdivided into smaller parts; a composite attribute is sometimes referred to as a molecular attribute.

Stored attribute	An attribute whose value is stored in the database.
Derived attribute	An attribute whose value can be calculated from one or more other attributes and thus is not stored in the database.
Multi-valued attribute	An attribute that can have more than one value for a particular entity.
Single-valued attribute	An attribute that can have a single value for a particular entity.
Mandatory attribute	An attribute that must be assigned a value.
Optional attribute	An attribute that need not be assigned a value.
Complex attribute	A meaningful clustering of composite and/or multi-valued attributes.
Semantic integrity constraint	A business rule that cannot be expressed explicitly or implicitly in the schema of a data model and is carried forward through the data modeling tiers in textual form.
Domain constraint	A data integrity constraint that establishes the range of acceptable values for an attribute.
Uniqueness constraint	A data integrity constraint that requires entities of an entity type be uniquely identifiable.
Unique identifier	An atomic or composite attribute whose values are distinct for each entity in the entity set.
Key attribute	An attribute that is a constituent part of a unique identifier.
Non-key attribute	An attribute that is not a constituent part of a unique identifier.
Relationship type	A meaningful association among entity types.
Degree of a relationship type	The number of entity types participating in the relationship type.
Binary relationship type	A relationship type in which two entity types are involved.
Ternary relationship type	A relationship type in which three entity types are involved.
Quaternary relationship type	A relationship type in which four entity types are involved.
Recursive relationship type	A relationship type in which one entity type is involved.
Relationship instance	An occurrence of a relationship type.
Relationship set	The set of all relationship instances.

Instance diagram	A diagrammatic representation of the relationship among the instances of the participating entity types.
Role name	A term used to describe the participation of an entity type in a relationship type.
Structural constraints of a relationship type	The data integrity constraints pertaining to relationship types specified in an ER diagram.
Cardinality constraint	A data integrity constraint that specifies the maximum number of instances of an entity type that relate to a single instance of an associated entity type through a particular relationship type.
Participation constraint	A data integrity constraint for an entity type in a binary relationship based on whether, in order to exist, an entity of that entity type needs to be related to an entity of the other entity type through this relationship type.
Total (Mandatory) participation	Occurs when every entity must participate in at least one occurrence of a relationship type.
Partial (Optional) participation	Occurs when every entity is not required to participate in at least one occurrence of a relationship type.
Existence dependency	Occurs when total participation of an entity type in a relationship type exists.
Weak entity type	An entity type that does not have its own unique identifier.
Base entity type	An entity type where the entities have independent existence.
Partial key	An attribute, atomic or composite, in a weak entity type which, in conjunction with a unique identifier of the parent entity type in the identifying relationship type, uniquely identifies a weak entity.
Discriminator	A term sometimes used in place of the term partial key.
Identifying relationship	A relationship between a base entity type and a weak entity type.
Deletion constraint	In the context of the deletion of an entity from the parent entity type, requires specific action either in the parent entity set or in the child entity set in order to maintain consistency of the relationships in

	the database.
Restrict rule	A deletion rule where the deletion of a parent entity in a relationship is restricted if all child entities related to the parent in the relationship should not be deleted.
Cascade rule	A deletion rule where the deletion of a parent entity in a relationship also causes all child entities related to the parent in the relationship to be deleted.
Set null rule	A deletion rule where the deletion of a parent entity in a relationship allows all child entities related to the parent in the relationship to be retained but no longer referenced to the parent.
Set default rule	A deletion rule where the deletion of a parent entity in a relationship allows all child entities related to the parent in the relationship to be retained but no longer referenced to the parent but instead referenced to a predefined default parent.
Cluster entity type	An entity type emerging as a result of a grouping operation on a collection of entity types and relationship(s) among them; indicated in an entity-relationship diagram by a dotted rectangle.

Chapter 2 Solutions

1. What is the difference between the conceptual world and the real world? Is it possible for a conceptual model to represent reality in total? Why or why not?
   Answer. The real world consists of (a) tangible objects of an object type and (b) intangible objects of an object type. The conceptual world consists of representations, in the form of entity types, of these real world tangible and intangible objects. It is not possible for a conceptual model to represent the real world, but only to represent the modeler's view of the real world (which we call in Chapter 1 the Universe of Interest or portion of reality).

2. Use examples to distinguish between:
   a. An object type and an entity type – *An object type is a named collection of properties that sufficiently describes an actual distinct type of identity. An object type can be tangible (e.g., a vehicle) or intangible (e.g., project). An entity type is a conceptual representation of an object type (e.g., a drawing of a vehicle).*
   b. An object and an entity – *An object is an actual occurrence of an object type (e.g., a actual vehicle). An entity is an occurrence of an entity type (i.e., the representation of a vehicle in a VEHICLE data set).*

c. A property and an attribute – *A property is a characteristic of an object type, whereas an attribute is a characteristic of an entity type. For example, the measured weight of a person would be termed a property, whereas the weight of the same person as recorded as a numeric value in a database would be termed an attribute.*
d. An entity and an entity instance – *An entity and an entity instance are one in the same. Both refer to an occurrence of an entity type. For example, the record for Anna Li in a STUDENT data set would be termed an entity or entity instance.*
e. An association and a relationship – *Associations exist in the real world between object types (e.g., a specific faculty member advises a specific group of students). In the conceptual world, such associations are referred to as relationships between entity types.*
f. An object class and an entity class – *An object class is a generalization of different related object types that have shared properties. For example, the object types basketball player, baseball player, and football player can be said to belong to the object class athlete. An entity class is a generalization of different related entity types that have shared attributes.*

3. Describe various data types associated with attributes.
Answer. A variety of data types can be associated with attributes. Those mentioned in Chapter 2 follow. A numeric data type is used when an attribute's value can consist of positive and negative numbers and is often used in arithmetic operations. An alphabetic data type permits an attribute to consist of only letters and spaces, whereas an alphanumeric data type allows the value of an attribute to consist of text, numbers, and certain special characters. A logical data type is associated with an attribute whose value can be either true or false (e.g., the attribute ***taxable****). A date data type is used for storing a date (e.g., date hired or date of birth).*

4. What is the difference between a stored attribute and a derived attribute?
Answer. A stored attribute is one whose value is stored in the database. On the other hand, a derived attribute is one whose value can be calculated or derived from the values of other attributes. At the physical level, typically derived attributes are not stored in a database.

5. What would be the domain of the attribute **County_name** in the state of Texas?
Answer. The domain of the attribute ***County_name*** *would be the set of all county names in the state of Texas.*

6. Distinguish between a simple attribute, a single-valued attribute, a composite attribute, a multi-valued attribute, and a complex attribute. Develop an example similar to Figure 2.3 that illustrates the difference between each type of attribute.
Answer. An attribute that has a discrete factual value and cannot be meaningfully subdivided is called an atomic or simple attribute. A composite attribute, on the other hand, can be meaningfully subdivided into smaller subparts with independent meaning. A single-valued attribute has a single value for a particular entity, whereas a multi-valued attribute can have more than one value for a particular entity. Both simple and composite attributes can be single-valued or multi-valued. A complex attribute is a cluster of composite and multi-valued attributes.

7. What is a unique identifier of an entity type? Is it possible for there to be more than one unique identifier for an entity type?
Answer. An attribute, atomic (i.e., simple) or composite, whose values are distinct for each entity in the entity set, is the unique identifier of the entity type. Yes, there can be more than one unique identifier for an entity type.

8. What is the difference between a key attribute and a non-key attribute?
Answer. A key attribute is any attribute that is part of a unique identifier. A non-key attribute is one that is not a constituent part of any unique identifier.

9. Consider the EMPLOYEE entity type given below.
   a. List all key and non-key attributes.
      Key attributes – Fname, Minit, Lname, Name_tag. The attributes Fname, Minit, Lname, and Name_tag are key attributes because they are constituent parts of the composite identifier Name. Non-key attributes – Address, Salary, Gender, Date_hired, No_of_dependents; these five simple attributes have nothing to do with any identifier.
   b. What is (are) the unique identifier(s)?
      Emp# and Name; in other words, we have a unique identifier that is a simple (atomic) attribute and another that is a composite attribute.
   c. Which attribute(s) is (are) derived attributes?
      No_of_dependents. Note: We have seen students indicate that a person's name as a combination of his or her first name, middle initial, last name, and name tag, constitutes a derived attribute. While perhaps a reasonable thing to do, this is not correct.
   d. Using the figure in Exercise 9 as a guide, develop sample data for four employees that illustrate the nature of the various mandatory and optional attributes in the EMPLOYEE entity. Be sure to illustrate the various ways the **Name** attribute might appear.

Emp#	Name_Fname	Name_Minit	Name_Lname	Name_tag	Address	Salary	Gender	Date_hired	Tenure
001	John		Doe		123 Any		Male	1/23/01	
002	Fred	J	Flinstone		321 There	39,000	Male	2/26/02	2
003	Barney		Rubble	BQT	322 There		Male	3/1/01	1
004	Lisa	M	Presley	LMP	2114 Pea	88,000	Female	4/2/02	

Note how this example shows how the Name attribute might appear four different ways: one time with just the first name and last name; one time with the first name, middle initial, last name; a third time with a first name, last name, and name tag; and a fourth time with a first name, middle initial, last name, and name tag. Since first name and last name are required attributes, they appear for each employee. Middle initial appears twice: once with a name tag and once without a name tag. Likewise, middle initial does not appear two times, once with a name tag and once without a name tag.

10. Discuss how to distinguish between an entity type and an attribute.
    Answer. An entity type corresponds to an aggregation of attributes.

11. Give an example of three entity types and accompanying attributes that might be associated with a database for a car rental agency.
    Answer.
    VEHICLE (VehicleID, LicensePlate#, Make, Model, Year, Mileage, …)
    RENTER (License#, Name, Address, City, State, Zipcode, Insurance, etc.)
    VEHICLE_RENTAL (VehicleID, License#, Dateout, Datein, Mileageout, Mileagein)

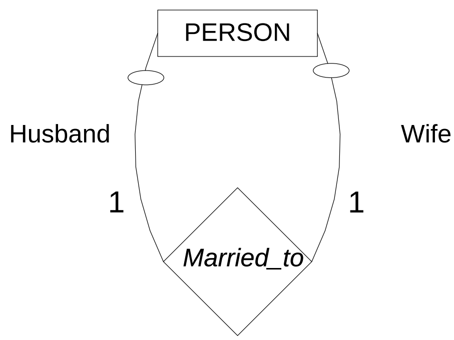
12. What is a relationship type? How does a relationship type differ from a relationship instance?
    Answer. A relationship type is a meaningful association among entity types. A relationship instance is an occurrence of a relationship type.

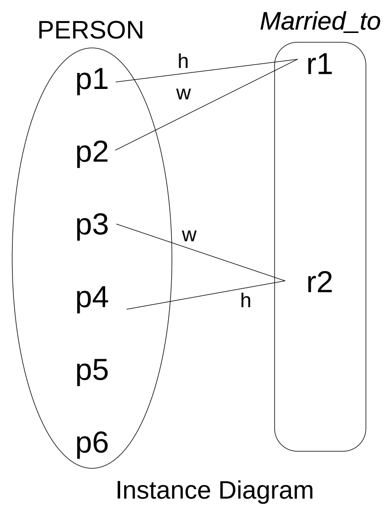
13. What is meant by the "degree" of a relationship?
    Answer. The degree of a relationship type is defined as the number of entity types participating in that relationship type.

14. What is the value of using role names to describe the participation of an entity type in a relationship type?
    Answer. Role names can be used to clarify the nature of the participation of each entity type involved in a relationship type. They can be particularly helpful in clarifying the nature of the structural constraints in a recursive relationship.

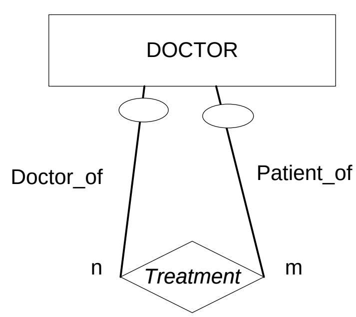
15. What is the difference between a binary relationship that exhibits a 1:1 cardinality constraint and a binary relationship that exhibits a 1:n cardinality constraint?
    Answer. In a binary relationship that exhibits a 1:1 cardinality ratio, each occurrence of entity type E1 is associated with at most one occurrence of entity type E2 and vice versa. In a binary relationship that exhibits a 1:n relationship, each occurrence of entity type E1 is associated with, at once, n occurrences of entity type E2, while each occurrence of entity type E2 is associated with, at most, one occurrence of entity type E1.

16. Describe how *Married_to* can be modeled as a recursive relationship.
Answer. Married_to can be modeled as a recursive relationship type if you think of it in terms of a PERSON entity type (e.g., Person 1 can be married to at most one other person, Person 2 can be married to at most one other person – either Person 1 or Person 3 or Person n, etc.). The following is an example of what an ER diagram and accompanying instance diagram might look like.





17. Create an example of a recursive relationship with an m:n cardinality constraint.
Answer. A possible example is a doctor treating other doctors as his/her patients and at the same time, a doctor being treated as a patient by other doctors.



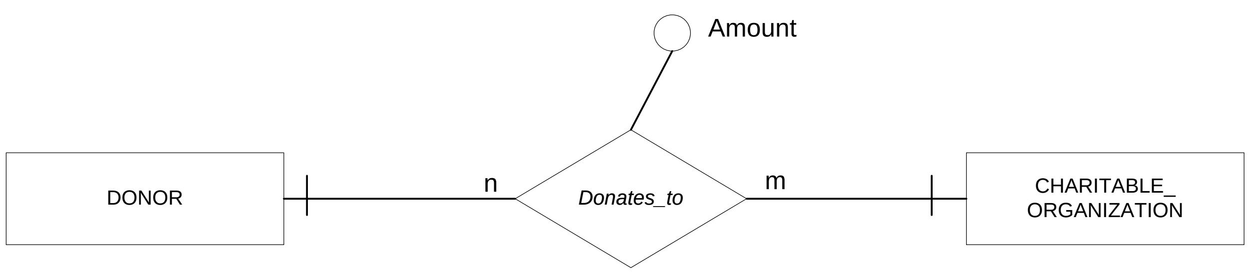
18. Distinguish between a participation constraint and minimum cardinality.
Answer. The participation constraint is also referred to as minimum cardinality.

19. Why can total participation of an entity type in a relationship type also be referred to as existence dependency of that entity type in that relationship type?
Answer. Total participation occurs if in order to exist, an entity type must participate in the relationship. Thus, total participation of an entity type in a relationship type can be termed existence dependency of that entity type in that relationship type.

20. How do cardinality constraints and participation constraints relate to the notions of total and partial participation?
Answer. A cardinality constraint has nothing to do with the notions of total and partial participation. Partial participation occurs if an entity type can exist without participation in a relationship, whereas total participation requires that an entity type must participate in the relationship in order to exist.

21. Discuss the difference between existence dependency and identification dependency.
Answer. Total participation of an entity type in a relationship type is defined as existence dependency. Identification dependency is existence dependency where a weak entity type is always dependent on the unique identifier of its identifying parent for its unique identification.

22. Give an example of a relationship type between two entity types where an attribute can be assigned to the relationship type instead of to one of the two entity types.
Answer. A good example involves the situation in Exercises 26 and 27 below, where the attribute commission is defined as the commission received by an insurance agent for the sale of a policy to a client. Another example could involve a relationship such as the following:



23. What is the difference between a base entity type and a weak entity type?
Answer. A base entity type is one where each entity has independent existence (i.e., each entity is unique). A weak entity type is one where each entity does not have independent existence (duplicate entities may exist). A weak entity type is shown to make it clear that the entity type does not have a unique identifier.

24. Define the term partial key.
Answer. A partial key is an identifier of a weak entity type. By itself, it is not a "unique" identifier, but when combined with the unique identifier of the identifying parent of the weak entity type, it uniquely identifies weak entities.

25. This is a narrative about a small university in Kodai, CA. There are several colleges in the university. Each college has a name, location, and size. A college offers many courses over four college terms or quarters – Fall, Winter, Spring, and Summer – during which one or more of these courses are offered. Course#, Name, and credit hours describe a course. No two courses in any college have the same Course#; likewise, no two courses have the same Name. Terms are identified by year and quarter, and contain numbers. Courses are offered during every term. The college also has several instructors. Instructors teach; that is why they are called instructors. Often, not all instructors are scheduled to teach during all terms; but every term has some instructors teaching. Also, the same course is never taught by more than one instructor in a specific term. Further, instructors are capable of teaching a variety of courses offered by the college. Instructors have a unique employee ID and their name, qualification, and experience are also recorded.
    a. List the business rules explicitly stated and implicitly indicated in the narrative.
    b. Study the narrative carefully and identify the missing information required for developing a semantically complete conceptual data model.

    a. *<u>Extracted business rules</u>:*

    - *There are four quarters; specifically, they are Fall, Winter, Spring, and Summer.*
    - *A college offers many courses.*
    - *At least 23 courses are offered during every quarter.*
    - *A college has several instructors.*
    - *An instructor need not teach in a given quarter.*
    - *An instructor is capable of teaching a variety of courses offered by the college.*

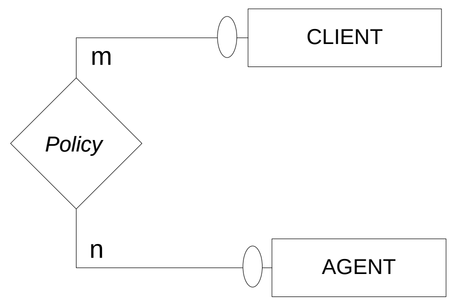
    <u>Inferred business rules</u>:

    - *An instructor must teach in some quarter.*
    - *An instructor must be capable of teaching at least one course that the college offers.*

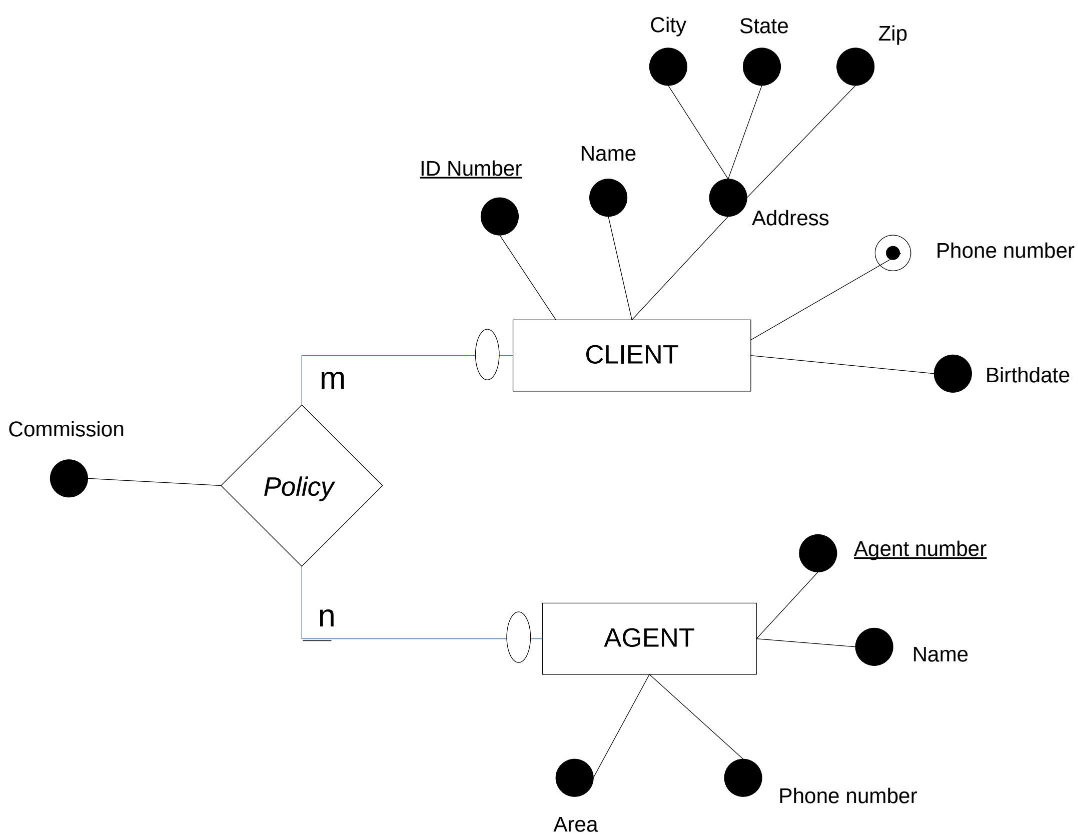
    b. *Ambiguities that require clarification:*

    - *Is a particular course offered in more than one quarter?*
    - *Are there courses that are just in the books but are never offered?*
    - *Can an instructor teach for more than one college in the university?*

26. The instance diagram shown below illustrates the relationship between Sullivan Insurance Agency's agents and clients. Using this instance diagram, write the narrative that describes the relationship between agents and clients. Your narrative should include a description of both the cardinality ratio and participation constraints implied in the instance diagram. In addition, draw the ER diagram that fully describes the relationship between the company's agents and clients.
Answer. Sullivan Insurance Company agents sell insurance policies to clients. More experienced agents often sell separate policies to many different clients, while newer agents may sell none. Likewise, not all clients have purchased a policy from an agent, whereas some clients have purchased a policy from more than one agent.

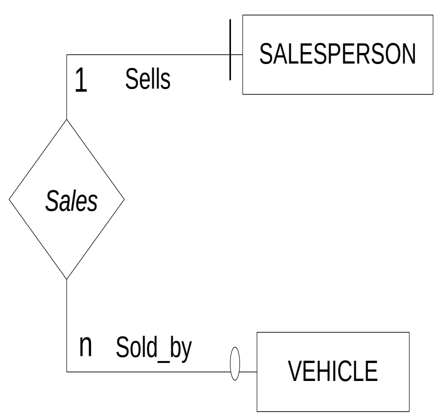


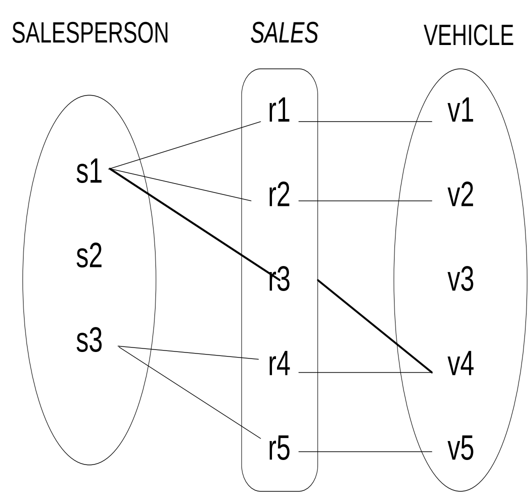
27. Revise the ER diagram drawn in the previous exercise to include the following mandatory attributes: CLIENT– ID number, name, address (city, state, zip), phone number(s), birthdate; AGENT – agent number, name, phone number, area; and commission received by an agent for selling a *Policy* to a client.
Answer. On this exercise, what we were really looking for was how the student handled the commission attribute. Note that commission is defined as "commission received by an agent for selling a Policy to a client. As a product of the relationship between a CLIENT and an AGENT, commission should be shown as an attribute of the relationship type Policy. Since the cardinality ratio is m:n, it cannot be shown as an attribute of agent. However, had commission been defined as simply an annual percentage of sales that an agent receives for a given period based on performance, then it would be an attribute of AGENT.



28. Use the instance diagram depicting the ternary relationship *Orders* shown on the next page to answer the following questions.
    a. Which customers order pens from the Galveston warehouse? *No customers order pens from the Galveston warehouse, as the Galveston warehouse does not supply pens.*
    b. Which items are ordered by customers from both warehouses? *Pencils are ordered from both warehouses.*
    c. Which warehouse fills one or more orders of items from both customers? *The Galveston warehouse takes one or more orders from both customers.*
    d. Describe orders filled from both warehouses. *This question is poorly worded and may generate several different answers. What we were looking for related to order r2, which involves Ives ordering pencils from both the Galveston and Charlotte warehouses. Some students may provide a description of all orders, which in essence contains what we were looking for along with a description of orders that technically we were not looking for.*
    e. What changes must be made to the instance diagram for order r1 to involve both pencils and pens? *Just as we have two warehouses going to order r2, this question necessitates two items leading into order r1. This was another question that perhaps could be worded better, as it can generate interesting answers to a slightly different question.*

29. The following two ER diagrams contain both a cardinality ratio constraint and a participation constraint.
    a. In the first ER diagram, is the instance diagram on the right consistent with the ER diagram on the left? Why or why not?

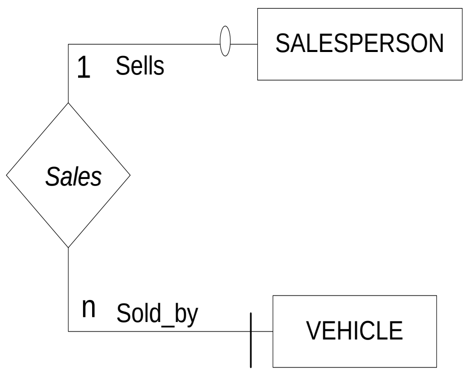


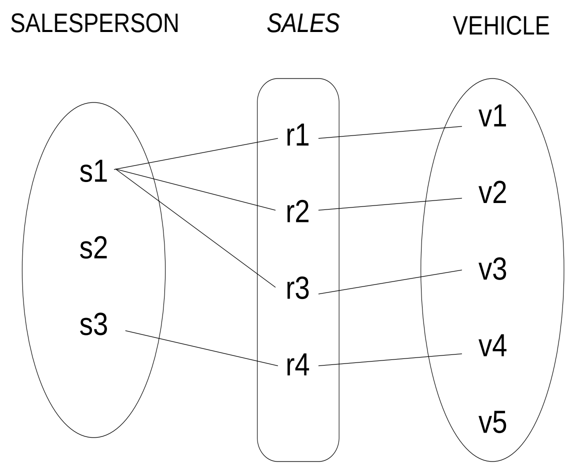


Instance Diagram

In the first ER diagram, the instance diagram on the right is not consistent with the ER diagram on the left. The ER diagram specifies total participation of VEHICLE in relationship *Sales*. But then, v3 in the instance diagram does not participate in the relationship.

b. In the second ER diagram, is the instance diagram on the right consistent with the ER diagram on the left? Why or why not?

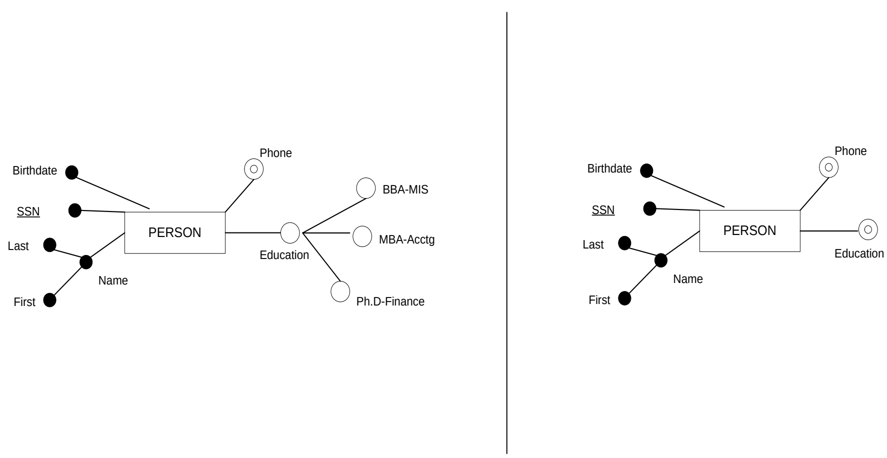




Instance Diagram

In the second ER diagram, the instance diagram on the right is not consistent with the ER diagram on the left. The ER diagram specifies total participation of SALESPERSON in relationship *Sales*. But then, s2 in the instance diagram does not participate in the relationship.

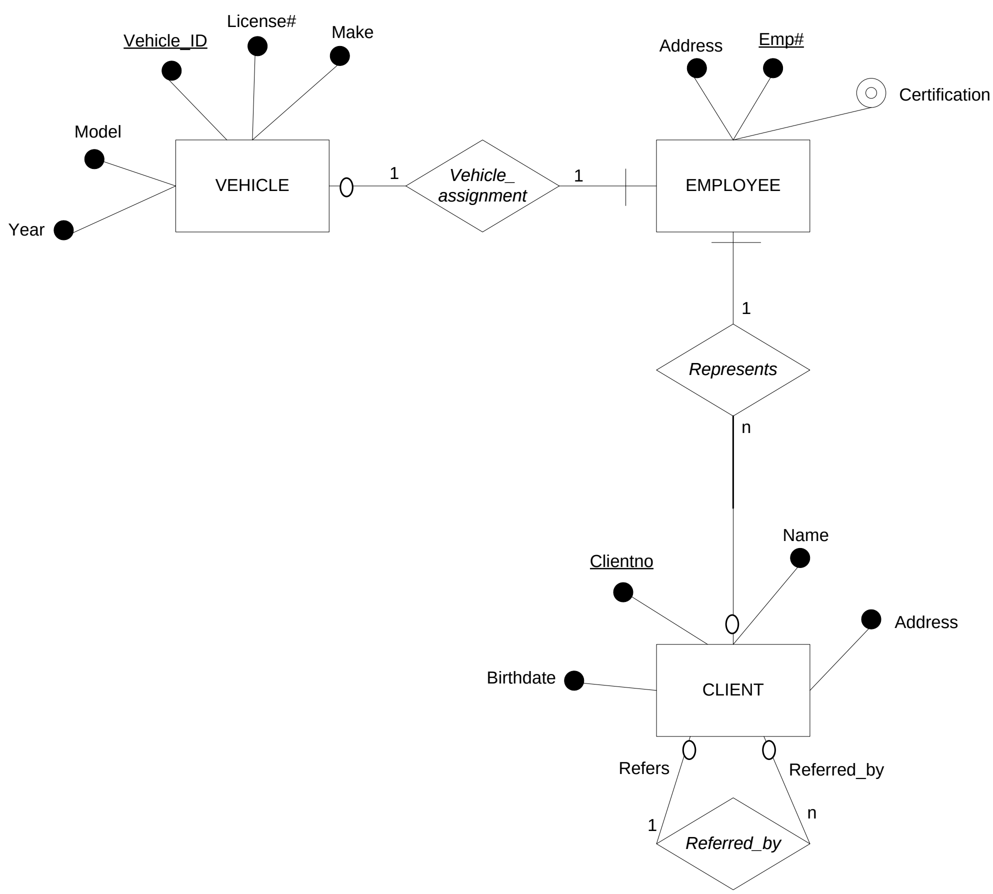
30. Suppose you want to show that a person can have multiple degrees. Would each of the following two ER diagrams get the job done? Why or why not? What is the difference?



The ER diagram on the left permits specification of a maximum of three degrees - in fact, only three specific degrees. The ER diagram on the right, however, is more flexible in that it permits specification of multiple degrees of any kind including no degrees.

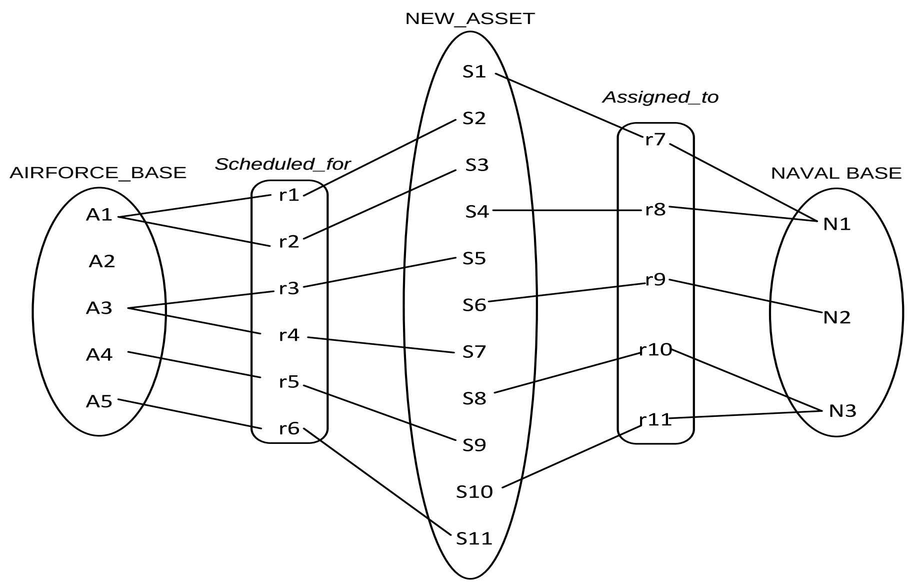
31. Adams, Ives, and Scott Incorporated is an agency that specializes in representing clients in the fields of sports and entertainment. Given the nature of the business, some employees are given a company car to drive, and each company car must be assigned to an employee. Each employee has a unique employee number, plus an address and set of certifications. Not all employees have earned one or more certifications. Company cars are identified by their vehicle id, and also contain a license plate number, make, model, and year. Employees represent clients. Not all employees represent clients, while some employees represent many clients. Each client is represented by one and only one employee. Sometimes clients refer one another to use Adams, Ives, and Scott to represent them. A given client can refer one or more other clients. A client may or may not have been referred to Adams, Ives, and Scott by another client, but a client may be referred by only one other client. Each client is assigned a unique client number. Additional attributes recorded for each client are name, address, and date of birth.

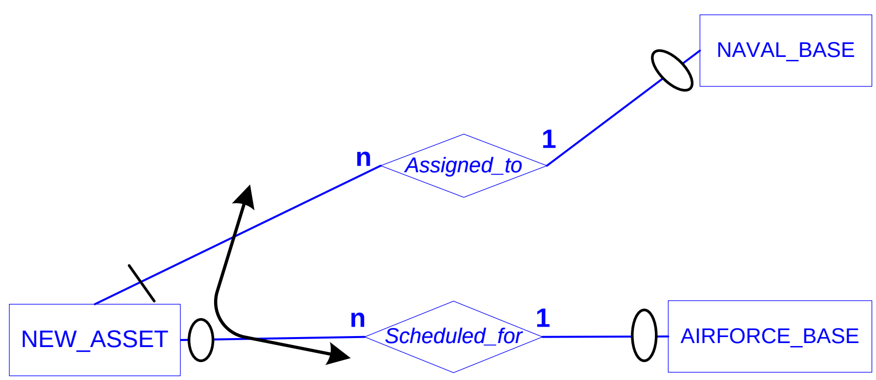
    Draw an ER diagram that shows the entity types and relationship types for Adams, Ives, and Scott. While you must name each relationship type and define its structural constraints, it is not necessary that you supply role names.

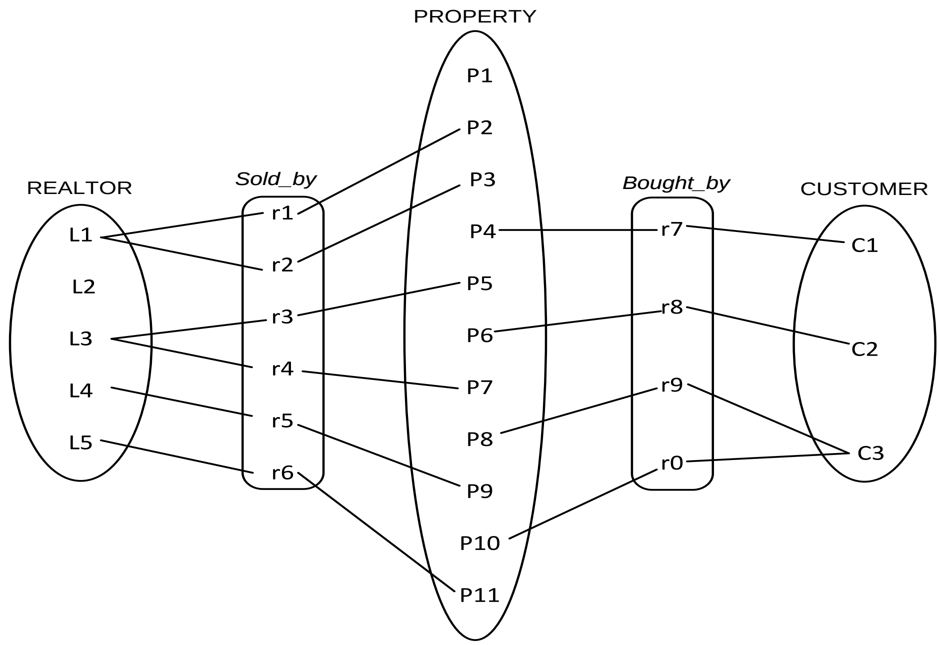


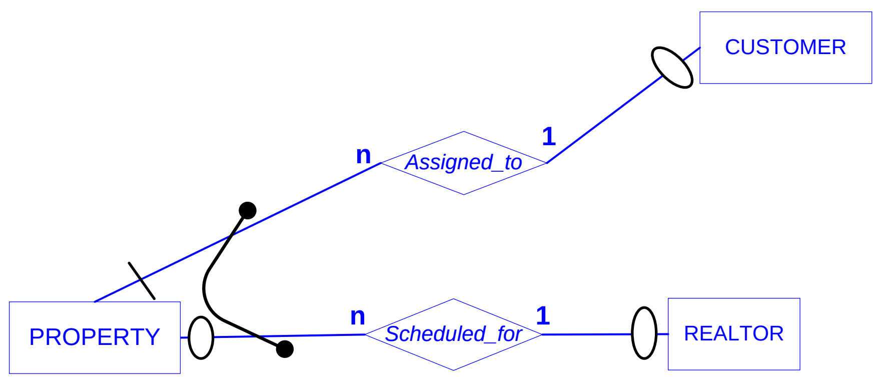
Comments: It is possible that some students may model CERTIFICATION as an entity type instead of a multi-valued attribute and then establish a relationship with an m:n cardinality ratio between EMPLOYEE and CERTIFICATION. This was a good alternative, although in doing so, it is necessary to invent an attribute for the CERTIFICATION entity type since none was given in the problem.

32. Draw the ER diagram for the two instance diagrams depicted here.









33. This vignette is a small excerpt from a comprehensive case about a clinic. Various physicians and surgeons working for a clinic are on an annual salary [o]. These doctors are identified by their respective employee numbers. The other descriptors of a doctor are: name, gender, address, and phone. Each physician's specialty and rank [o] are captured; each surgeon's, specialty and skill are also captured; a surgeon may have one or more skills.

    Every physician serves as a primary care physician for at least seven patients; however, no more than 20 patients are allotted to a physician. Every patient is assigned one physician for primary care. Some patients need surgeries; others don't. Surgeons perform surgeries for the patients in the clinic. Some do a lot of surgeries; others do just a few. The date and operation theater [o] for each surgery needs to be recorded, too. *Removal of a surgeon from the clinic database is prohibited if that* *surgeon is scheduled to perform any surgery. However, if a patient chooses to pull out of the surgery schedule, all surgeries scheduled for that patient are cancelled.*

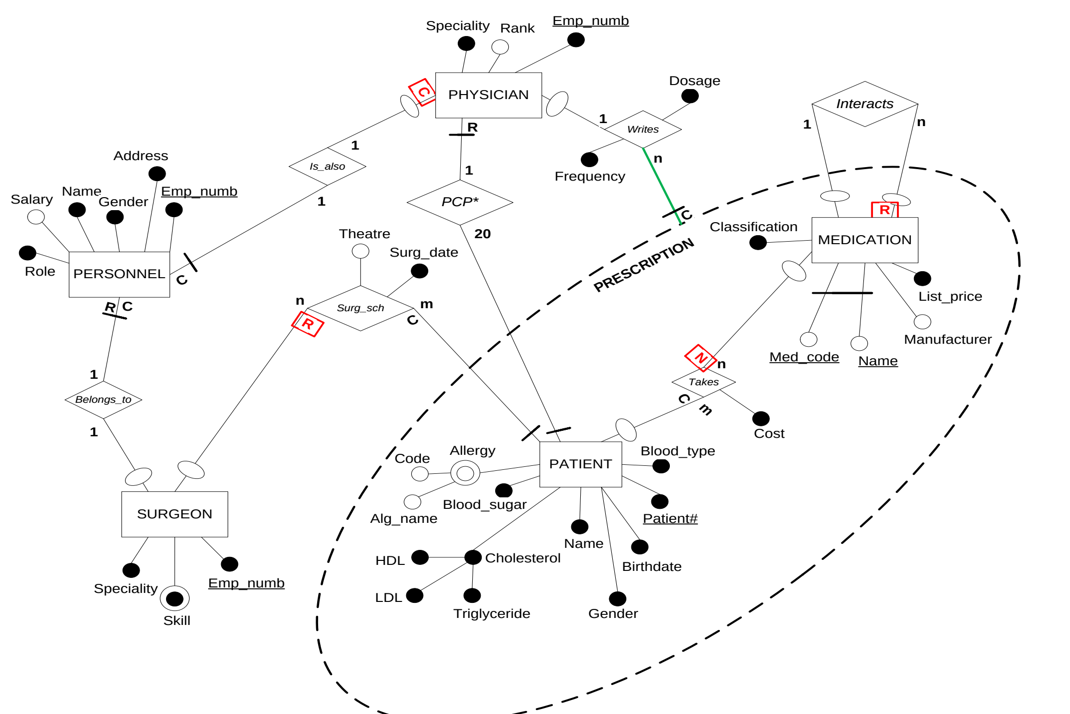
Data for patients include: patient number (the unique identifier of a patient), name, gender, date of birth, blood type, cholesterol (consisting of HDL, LDL, and triglyceride), blood sugar, and the code and name of allergies, if any.

Physicians may prescribe medications to patients; thus, it is necessary to capture which physician(s) prescribe(s)s what medication(s) to which patient(s) along with dosage and frequency. *In addition, no two physicians can prescribe the same medication to the same patient. If a physician leaves the clinic, all prescriptions written by that physician should also be removed because this information is retained in the archives.*

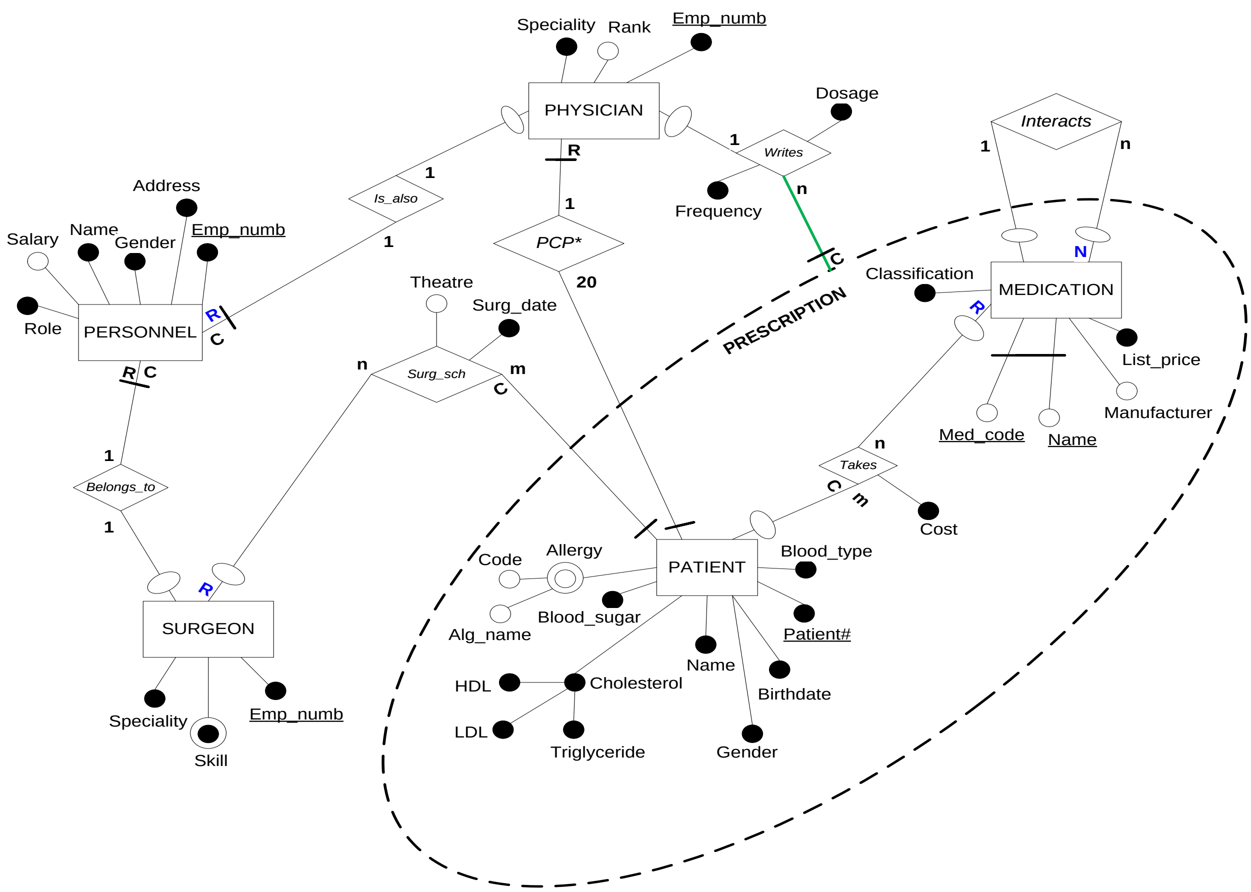
A patient may be taking several medications, and a particular medication may be taken by several patients. Despite its list price, a medication's cost varies from patient to patient, perhaps because of the difference in insurance coverage. The cost of a medication for a patient needs to be captured. A medication may interact with several other medications. *When a medication is removed from the system, its interaction with other medications, if any, should be voided. When a patient leaves the clinic, all the medication records for that patient are removed from the system.*

Medications are identified by either their unique medication codes or by their unique names. Other attributes of a medication are its classification, list price, and manufacturer [o]. For every medication, either the medication code or the medication name must be present—not necessarily both.

Note: [o] indicates optionality of value for the attribute. Develop an ER model for this scenario.



Deletion constraints enclosed in a box (highlighted in red) reflect erroneous/conflicting deletion (business) rules. A few different ways of correction are possible. Once such correction is shown in the next ER diagram


Speciality
Rank
Emp_numb
PHYSICIAN
R
1
Dosage
Writes
n
Frequency
C
Interacts
1
n
N
PRESCRIPTION
Classification
MEDICATION
R
List_price
Manufacturer
Med_code
Name
Is_also
1
1
PCP*
1
20
Address
Salary
Name
Gender
Emp_numb
Role
PERSONNEL
R
C
R
C
Theatre
Surg_date
n
Surg_sch
m
C
Takes
n
C
m
Cost
1
Belongs_to
1
R
SURGEON
Speciality
Skill
Emp_numb
Code
Allergy
PATIENT
Blood_type
Alg_name
Blood_sugar
Patient#
Name
HDL
Cholesterol
Birthdate
LDL
Triglyceride
Gender