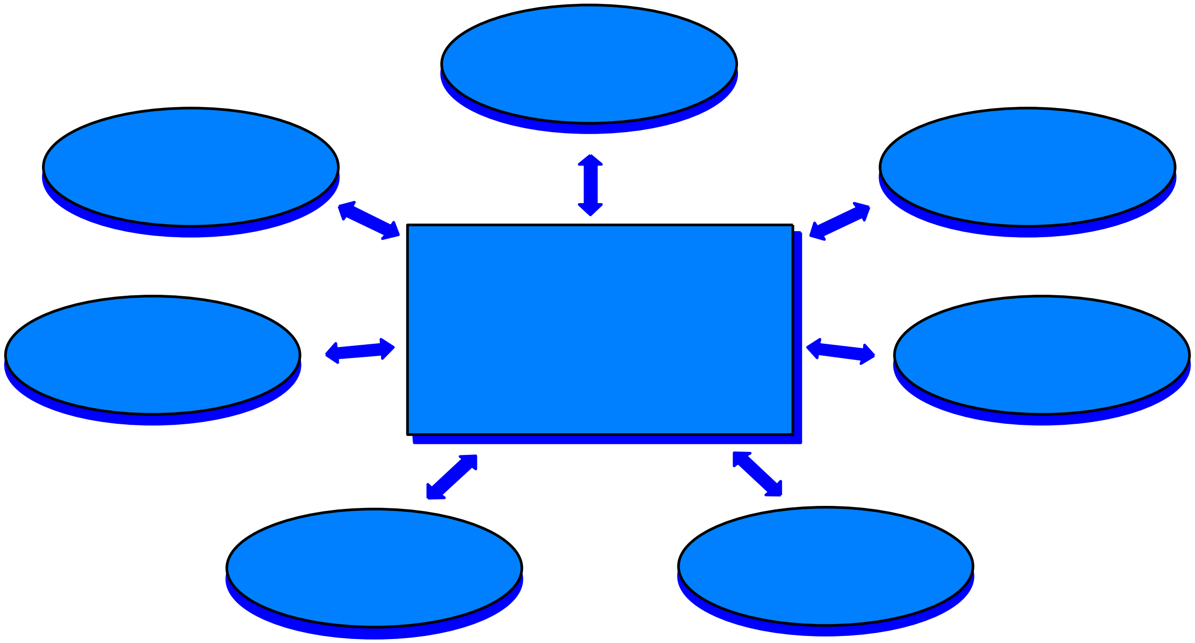


# DataBase Data Modeling Using the Entity-Relationship Model

# Data Modeling

- Process of creating a logical representation of the structure of the database
- The most important task in database development

# The Data Model

- A **data model** is a plan, or blueprint, for a **database design**.
- A data model is more generalized and abstract than a database design.
- It is easier to change a data model than it is to change a database design, so it is the appropriate place to work through conceptual database problems.

# The Entity Relationship Model

- The Entity-Relationship model is a set of concepts and graphical symbols that can be used to create conceptual schemas

- Developed by Peter Chen of M.I.T. in 1976 in a landmark paper, "The Entity-Relationship Model: Toward a Unified View of Data," ACM Transactions on Database Systems, Vol. 1, No. 1

- Chen's initial model has been refined over the years

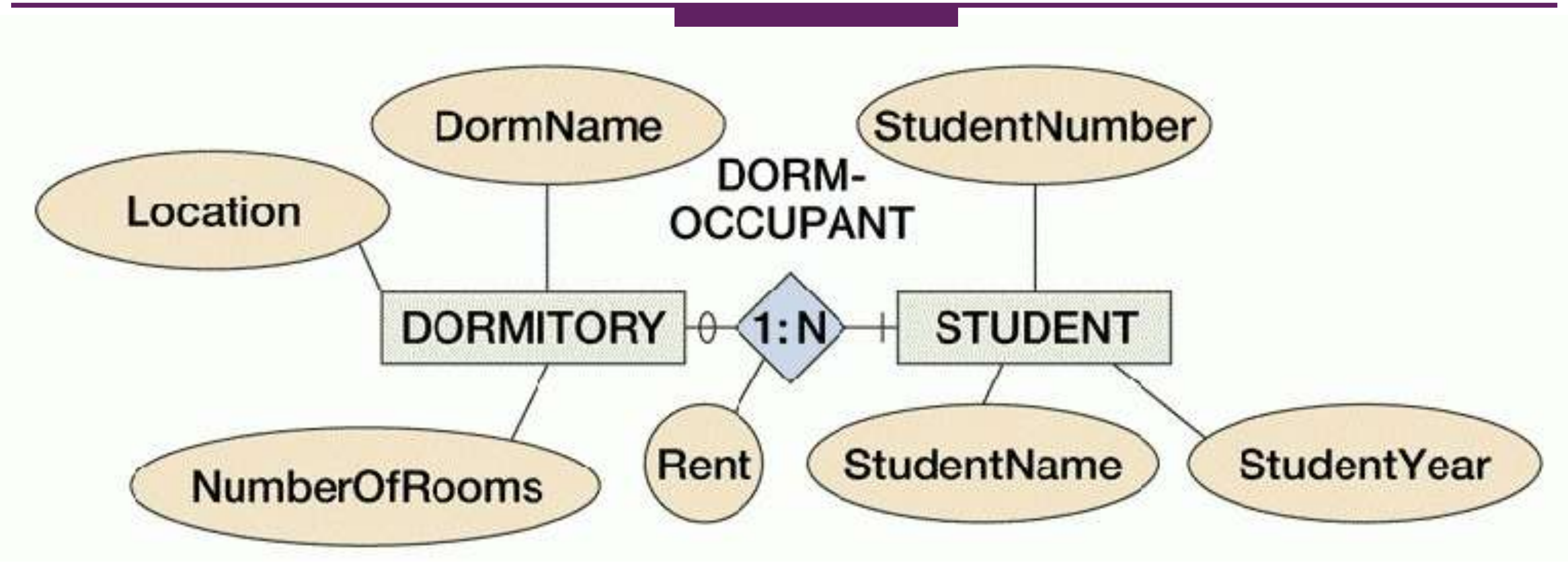
# Versions/Evolution of the E-R Model

- Original E-R model by Peter Chen (1976)
- Extended E-R model (1986): added subtypes, now the most widely used E-R model, and what we will use in IS431
- Information Engineering/IE model (also called the Crow’s Foot model) developed by James Martin in 1990
- IDEF1X (1994) : national standard by the National Institute of Standards and Technology
- Additionally:
  - Unified Modeling Language (UML) Style E-R Models
  - Semantic Object Data Models (SOMs)

# The Four Major Components of the Entity Relationship (E-R) Model

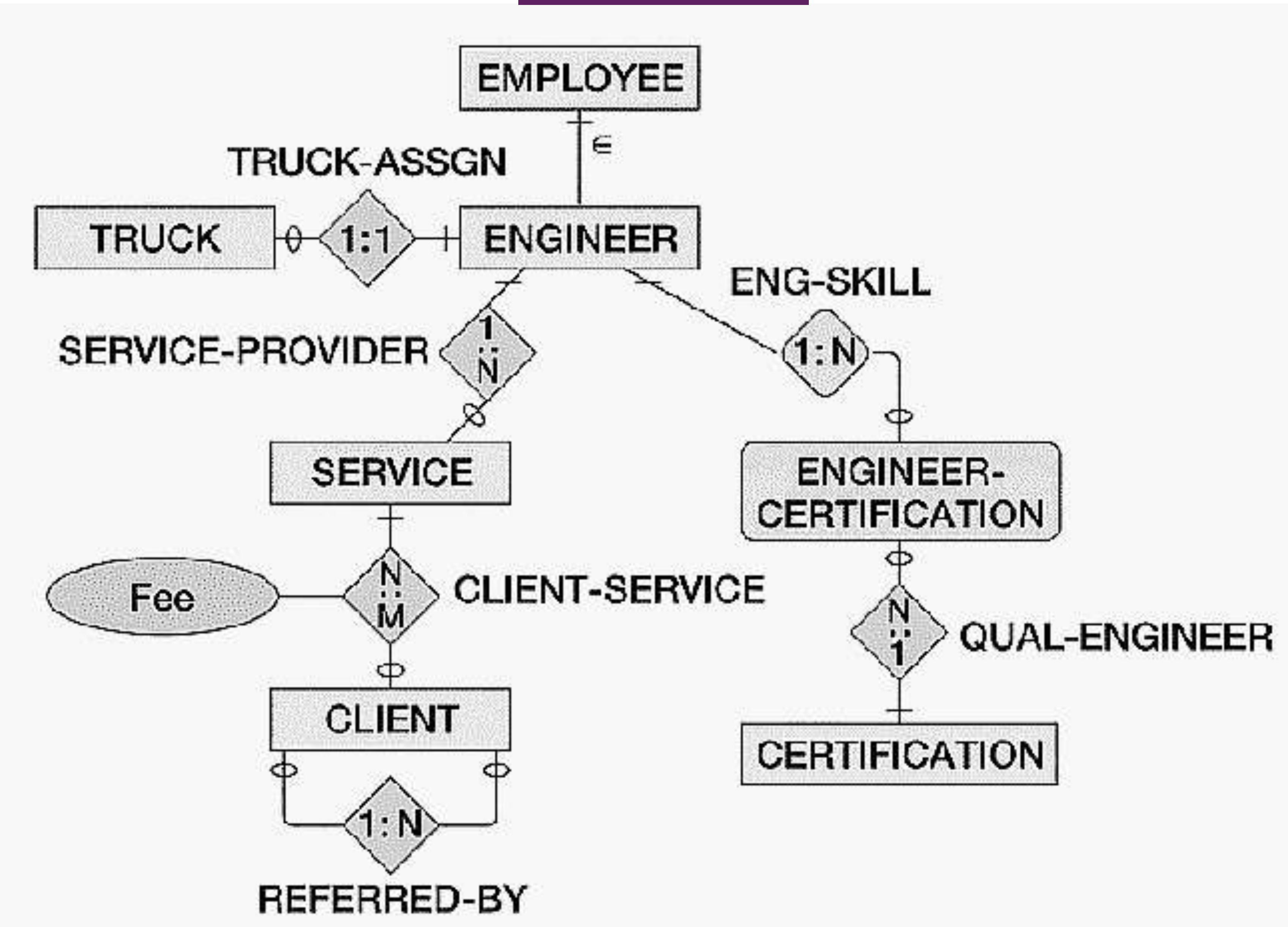
- Entities (really entity sets)
- Attributes
- Relationships
- Identifiers

# An Example of an E-R Diagram



The notation will be explained…

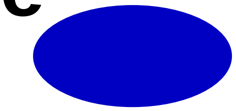
# An Extended E-R Model



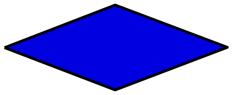
# The Geometric Symbols in an E-R Model

- Entities are represented by rectangles.

- Attributes are represented by ovals, that are connected to the entity by a straight line.



- Relationships are represented by diamond shaped symbols.

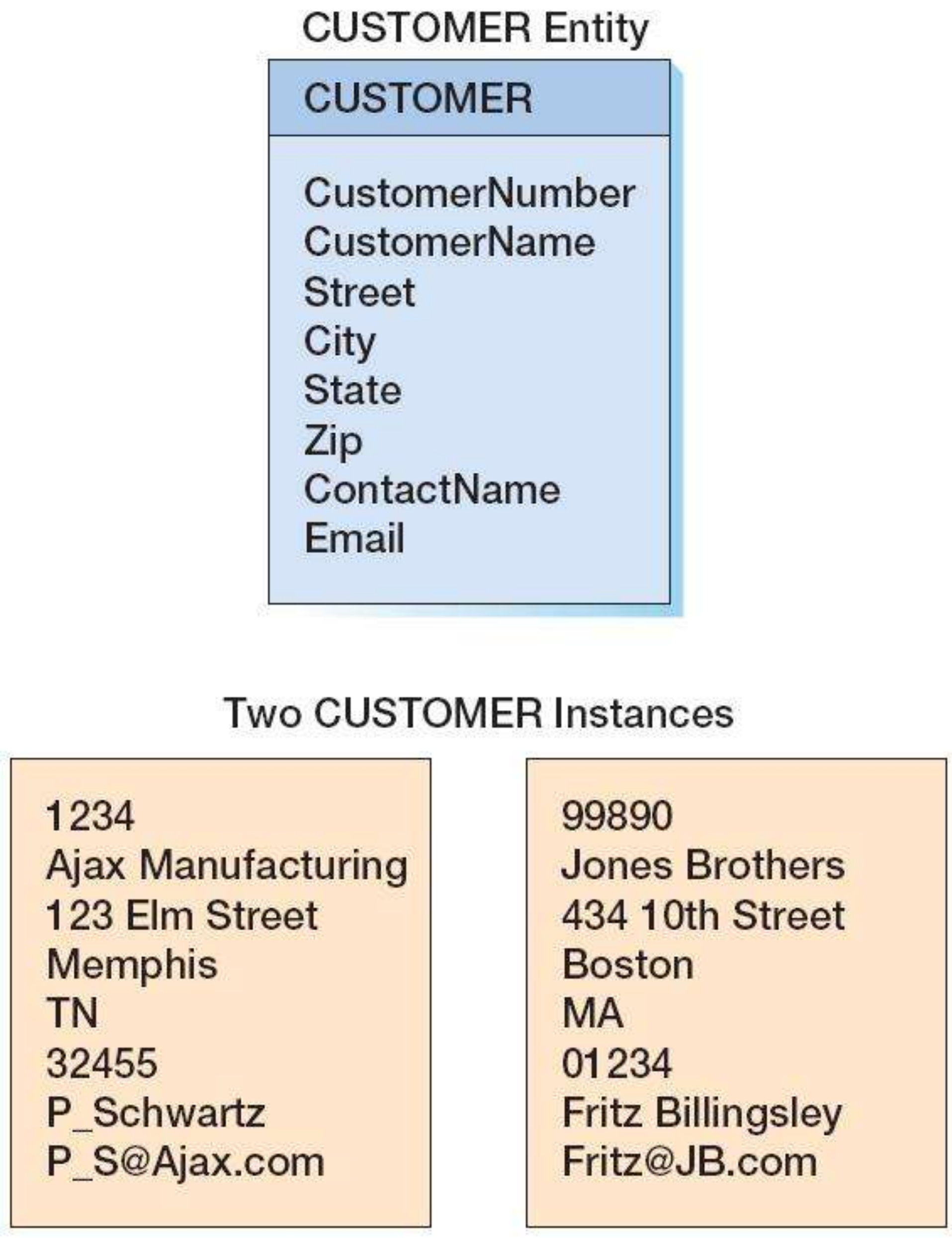


- The name of the entity (class) or attribute or relationship is usually placed inside the symbol used for that object. (Sometimes, with relationships, the name is placed adjacent.)

# Entities and Entity Sets

- **An entity set (or class) is something that can be identified and the users want to track**
  - **Entity class is a collection of entities described by the entity format in that class**
  - **Entity instance is the representation of a particular entity**
- **There are usually many instances of an entity in an entity class**
- **Consider an entity class STUDENT**
  - **An entity instance would be a particular student in the entity class, for example, Matthew Jones might be an entity instance**
- **A general notational convention is to CAPITALIZE the entity set's name**

# Example: An Entity Set CUSTOMER and Two Entity Instances



CUSTOMER Entity

CUSTOMER

CustomerNumber
CustomerName
Street
City
State
Zip
ContactName
Email

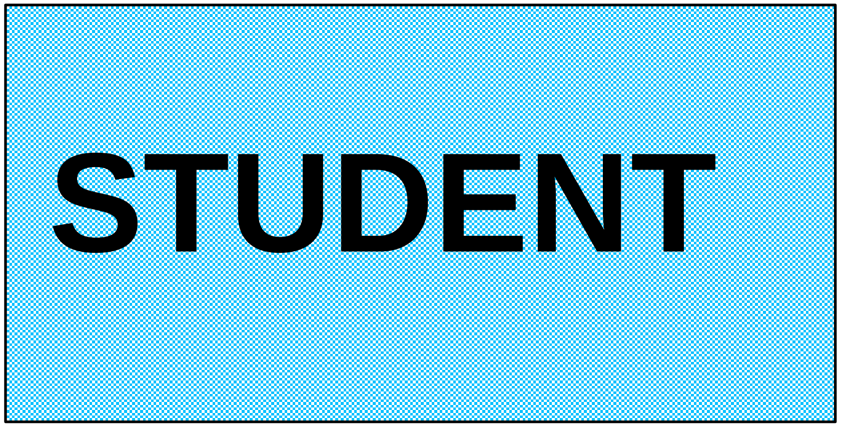
Two CUSTOMER Instances

1234
Ajax Manufacturing
123 Elm Street
Memphis
TN
32455
P_Schwartz
P_S@Ajax.com

99890
Jones Brothers
434 10th Street
Boston
MA
01234
Fritz Billingsley
Fritz@JB.com

# Representing Entities (Entity Sets) in an E-R Model

- An entity set/class is represented by a rectangle with the name of the entity set/class in the rectangle.



# Attributes

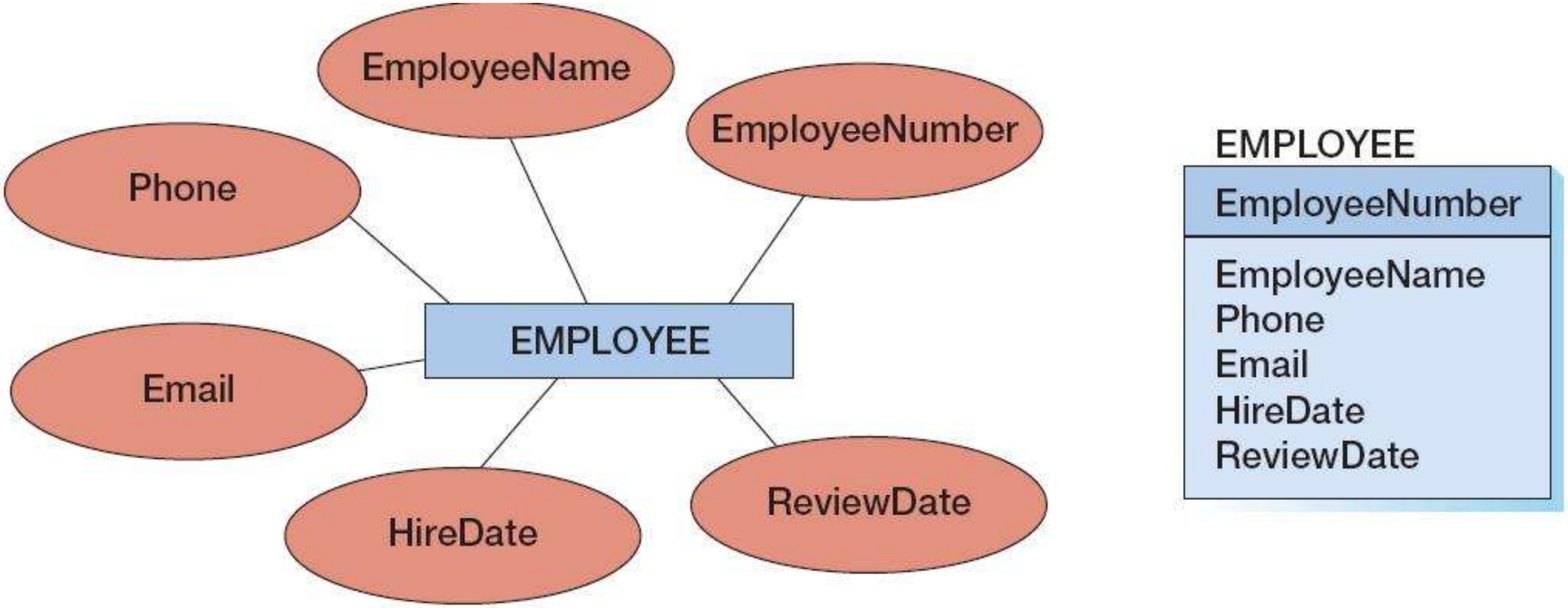
- Description of the entity’s characteristics
- All instances of a given entity class have the same attributes
  - – Composite attribute: attribute consisting of the group of attributes
  - – Multi-value attributes: attribute with more than one possible value

# Representing Entities and Their Attributes

- **Method 1**: Place the entity name in a rectangle, and the attributes as ovals/ellipses attached to the rectangle.

- **Method 2**: Place the entity name at the top of a portrait-shaped rectangle, and place the attributes below the entity name in a rectangle.

# Variations of Displaying Attributes

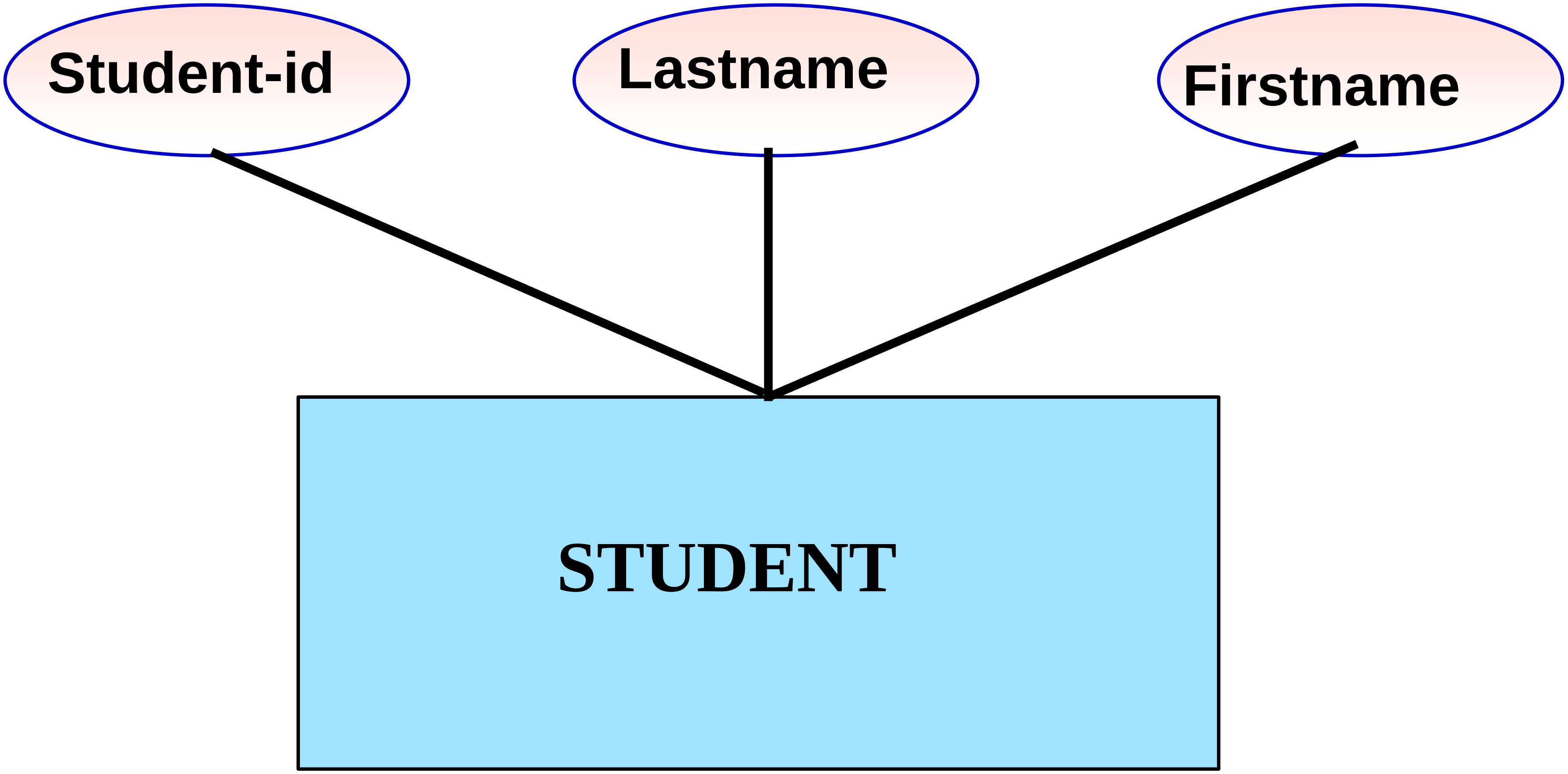


(a) Attributes in Ellipses

(b) Attributes in Rectangle

# A Student Entity Set With Three Attributes

**(Using Ovals/ellipses)**



**(Can you describe what the entity set STUDENT would look like, if we used Method 2)?**

# Review of Attributes

- **Attributes are the properties that define the entity's characteristics**
- **The E-R model assumes that all instances of a given entity set have the same attributes.**
  - ✓ *So, for the ER model in the prior slide for STUDENT, instances of STUDENT, such as Henry Gordon, Joyce Johnson, Jeffrey Chan, etc, will each have a Student-ID, a Lastname and a Firstname attribute.*

# Identifiers

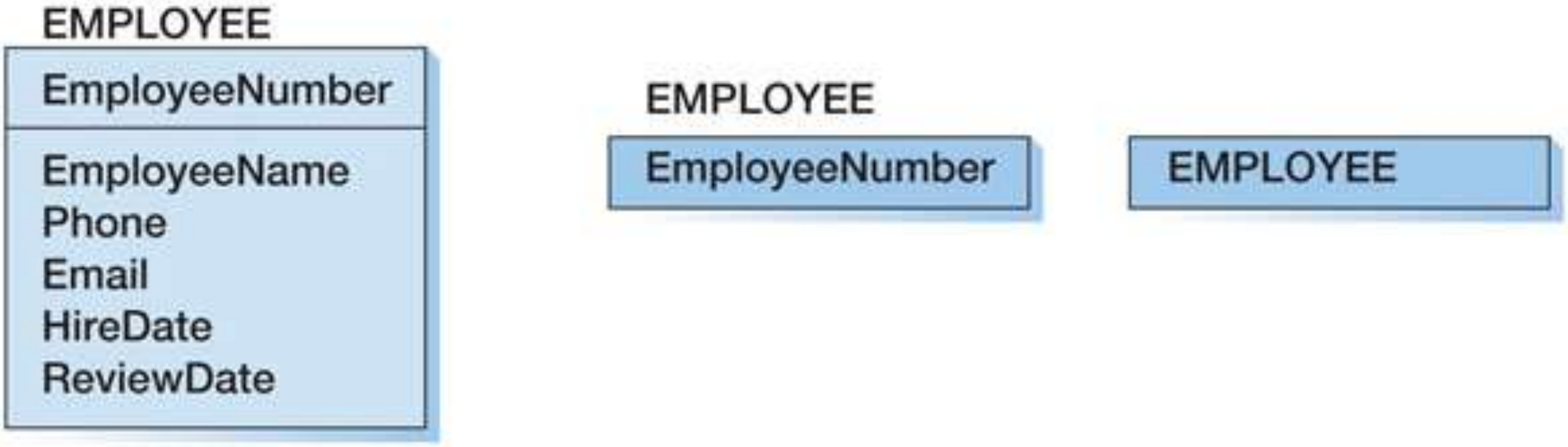
- **Identifiers are one or more attributes of entity instances which serve to name, or identify, the entity instance.**
  - ✓ *For the STUDENT entity set, identifiers are **Student-id** and the composite identifier (**Last-name, First-name**).*
    - ***Composite identifiers are identifiers that consist of two or more attributes***
  - ✓ *Identifiers are either **Unique** (identifies one and only one entity) or **Non-Unique** (identifies a set of entities).*
- **We often refer to a Unique Identifier as the Primary Key of a table (relation)**.

# Attributes & Domains

- Attributes will also have a domain.
  - The domain is the attribute's set of possible values.
  - The domain of the attribute "Grade Point Average" is a real number between 0 and 4.
  - The domain of the attribute "Gender" consists of only two possibilities, M or F (or some other equivalent code).
  - Attributes may share a domain.
    - The attribute PROFESSOR_AGE and STUDENT_AGE share the domain of all possible ages.

# Variations on Displaying Entity Sets and Attributes in an Entity Set

Depending on the size of our ER model, we might display the entity set name and ALL of the attributes, or the entity set name and the identifier, or just the entity set name:



(a) Entity with All Attributes

(b) Entity with Identifier Attribute Only

(c) Entity with No Attributes

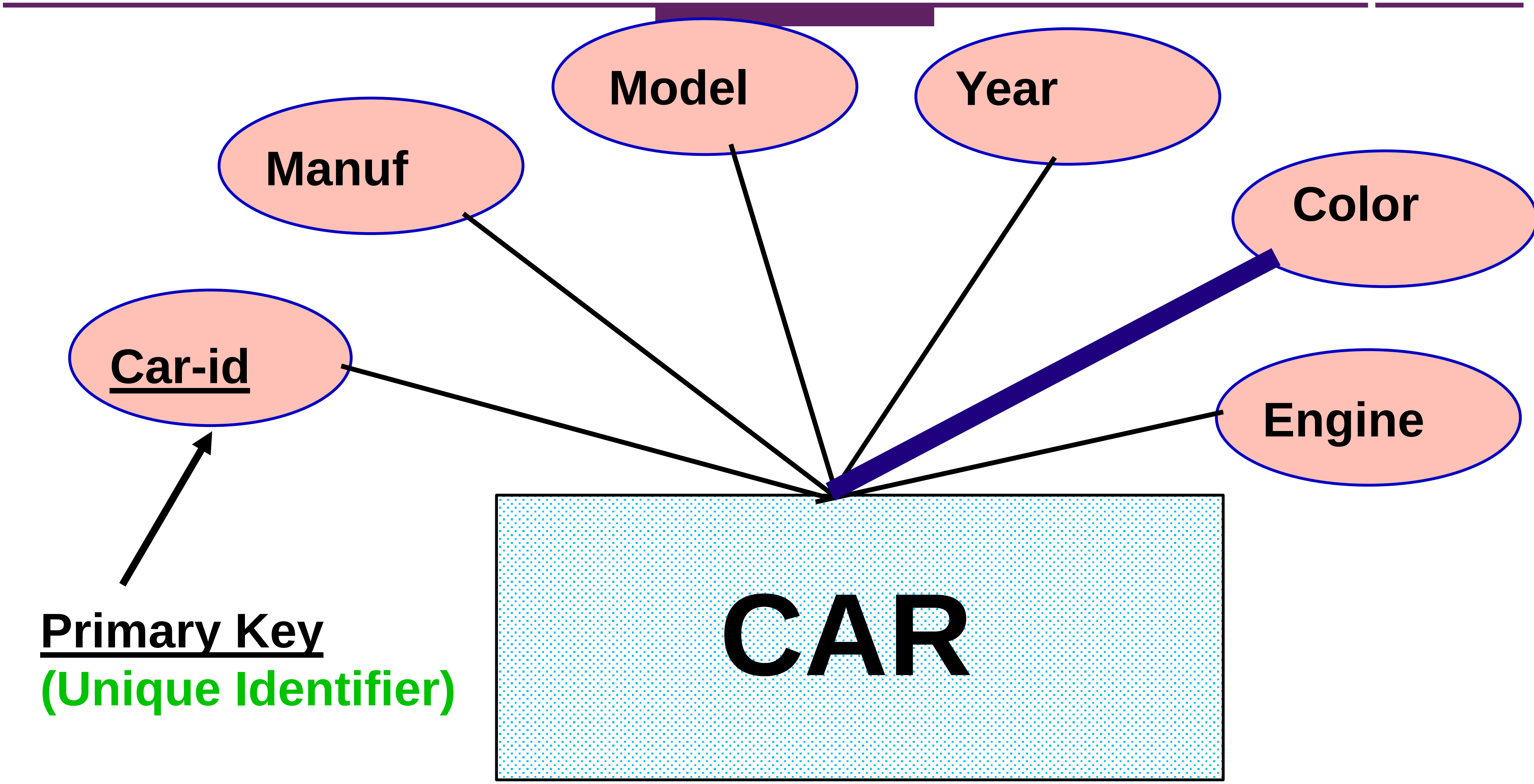
# Classification of Attributes By Allowable Values

- A Single-Valued Attribute can have only a single value.
  - Example: Age is a single valued attribute of the entity person, because a person can have only one age.
- A composite attribute is a logical grouping of single-valued attributes.
  - Example: The composite attribute Address consists of the group (Street, City, State, Zip).

# Classification of Attributes By Allowable Values - Continued

- A Multi-Valued Attribute can have many values.
  - ➔ *Example: PackageExpertise is a multiple valued attribute of the entity person, because a person may have expertise in many packages (WordPerfect, Excel, Photoshop, Access, Powerpoint, AutoCad, MathCad, etc.)*
    - ✓ A multi-valued attribute may also be composite. An example would be, assuming we allowed multiple phone numbers for an entity, Phone (area code, number)
- In an E-R model, Multiple Valued Attributes are shown by a double line (or a bold line) connecting the attribute to the entity (entity class).

# A Multi-Valued Attribute “Color”

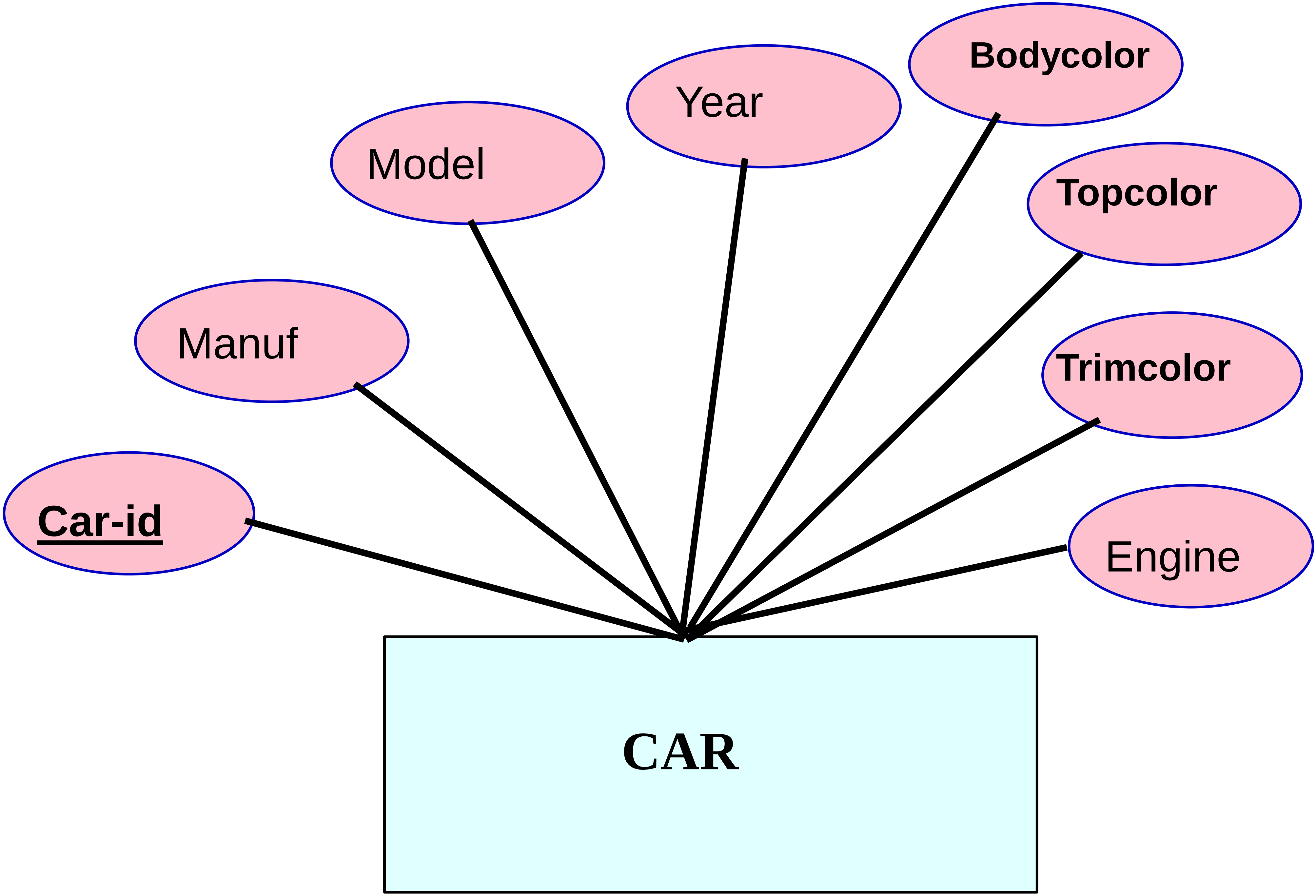


**Primary Key**

(Unique Identifier)

# Translating E-R models with Composite Attributes into a Relational DBMS

- For the original entity, create additional new attributes, one for each of the original composite attributes components.
  - ✓ Thus, the composite attribute Color will be split into new attributes: Topcolor, Bodycolor and Trimcolor, all linked to the entity class Car.

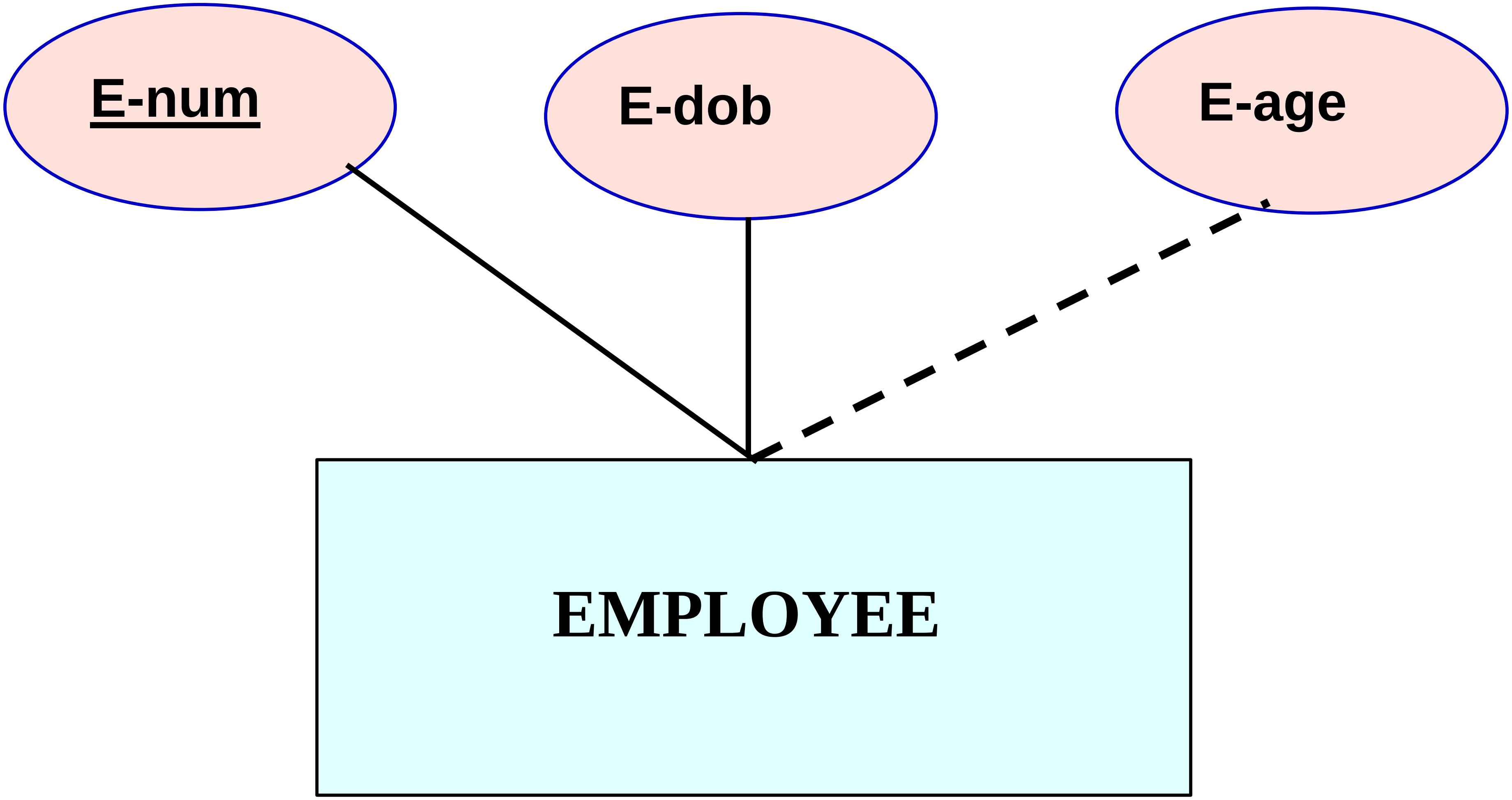


Splitting the Composite  Attribute into New Attributes

# Derived Attributes

- A Derived Attribute does not physically exist within the database, but is derived (computed) by an algorithm or computation.
  - Example: A person's AGE attribute can be derived by subtracting the date of birth (DOB) from the current date.
  - Example: Total cost can be derived by multiplying quantity ordered by unit price.
- A Derived Attribute is indicated in an E-R Model by a dotted line connecting the attribute to the entity.

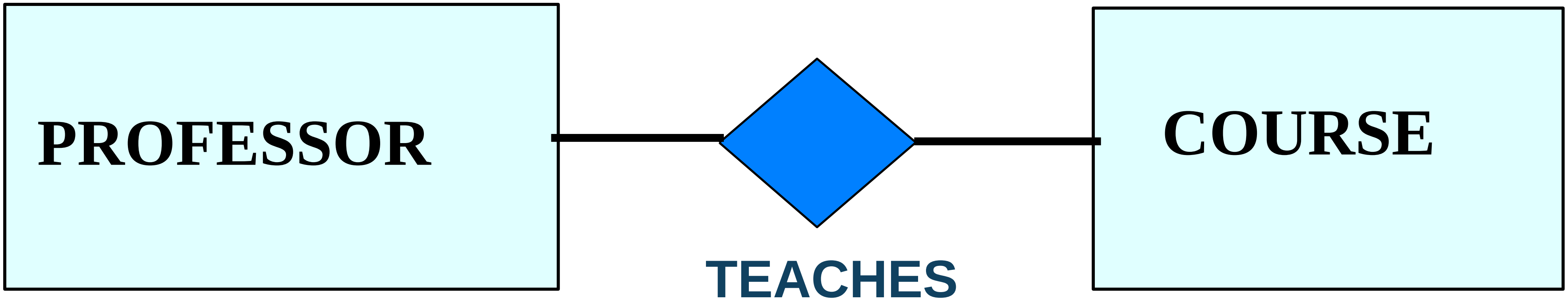
# A Derived Attribute



# Relationships in an E-R Model

- A relationship is an association between entities:
  - ✓Relationship classes: associations among entity classes
  - ✓Relationship instances: associations between entity instances
- Relationships are represented by diamond-shaped symbols, connecting the entities in the relationship (the entities are referred to as the *participants* in the relationship).
- In the original Entity-Relationship model. relationships were allowed to have attributes, but not in the extended E-R model today.

# A Relationship Class Between a Professor Entity Set/Class and a Course Entity Set/Class



What would be a relationship instance in this case?

# The DEGREE of a Relationship

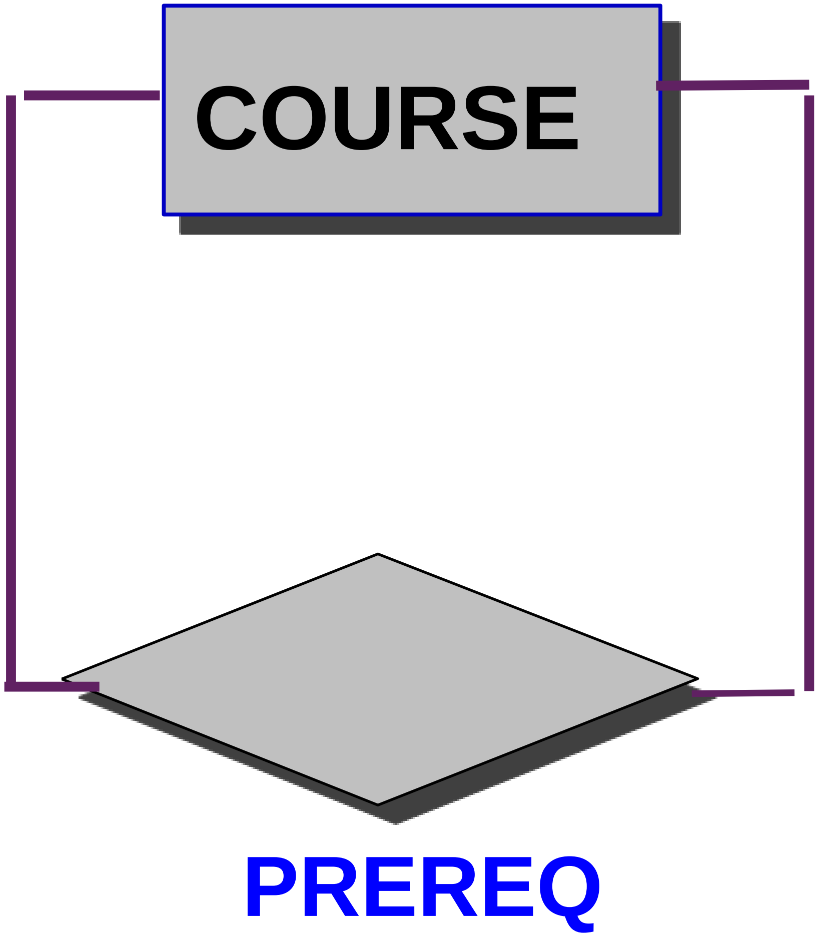
The **degree** of a relationship is the number of associated entity sets (participants) in the relationship.

✓A **UNARY RELATIONSHIP** exists when an association exists within a single entity

✓A **BINARY RELATIONSHIP** exists when two entities(participants) are in the relationship.

✓A **TERNARY RELATIONSHIP** exists when three entities (participants) are in the relationship.

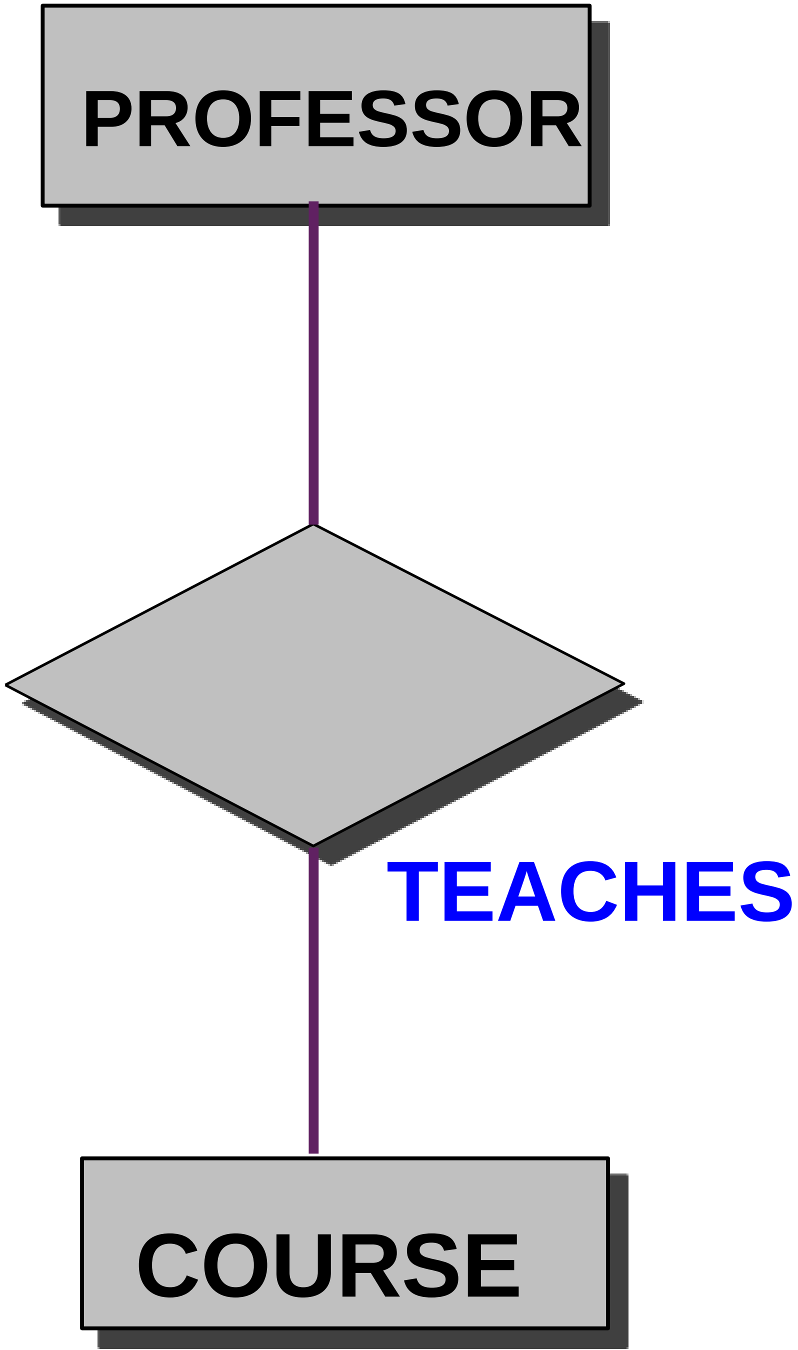
## UNARY



COURSE

PREREQ

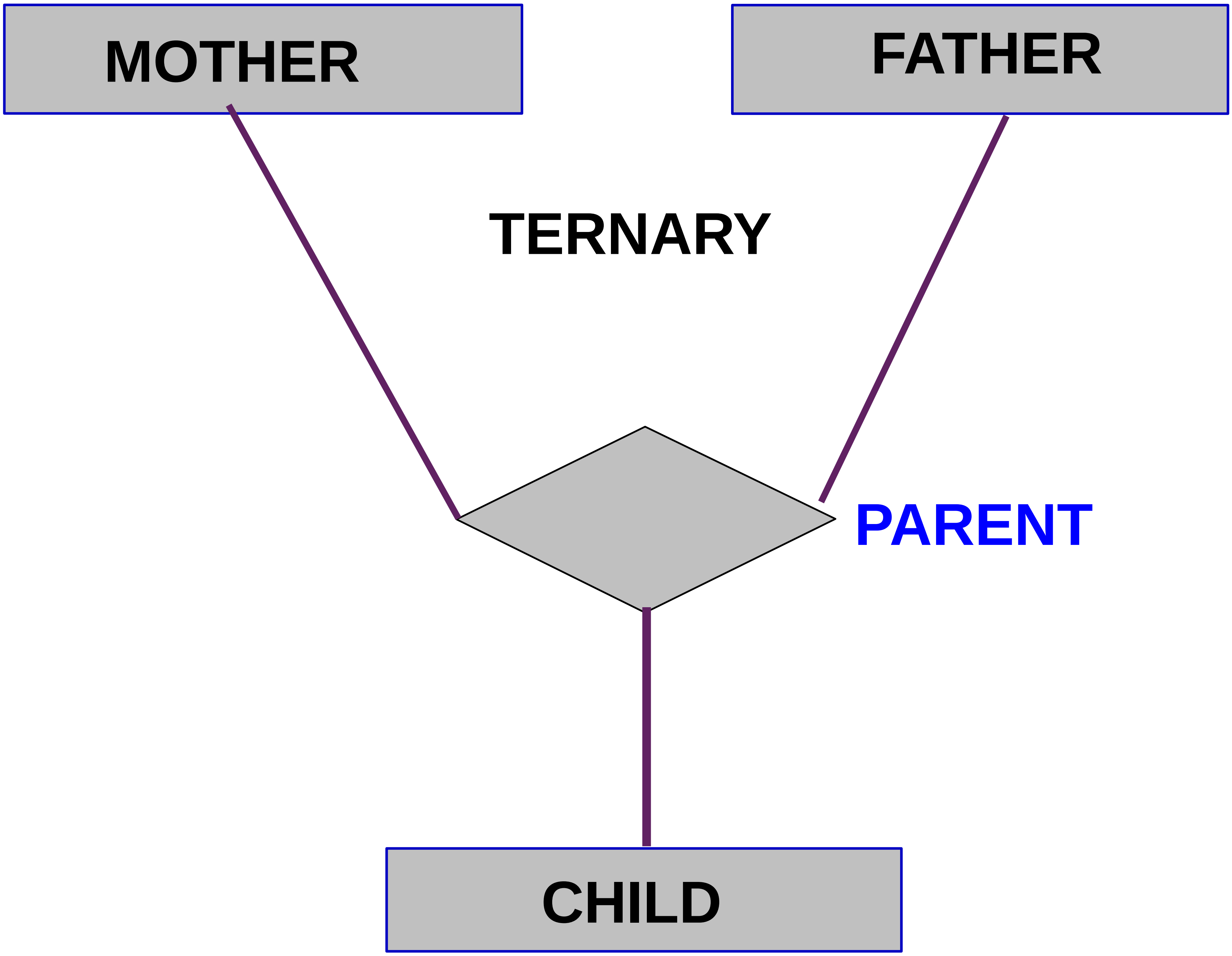
## BINARY



PROFESSOR

TEACHES

COURSE

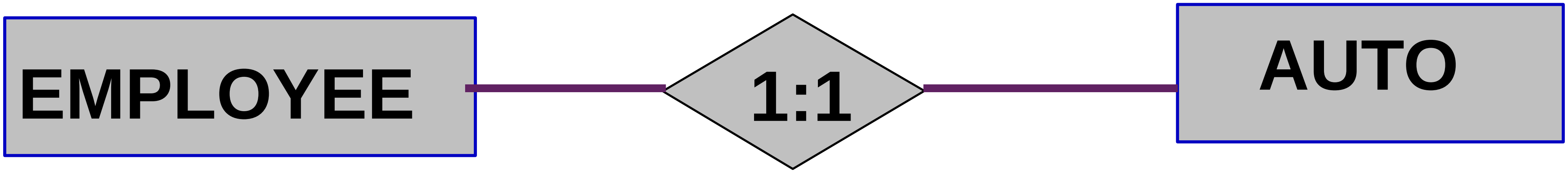

MOTHER
FATHER
TERNARY
PARENT
CHILD

# THREE TYPES OF BINARY RELATIONSHIPS

- One to One
  - Abbreviated as "1:1"

- One to Many
  - Abbreviated as "1:N"

- Many-to-Many
  - Abbreviated as "N:M"

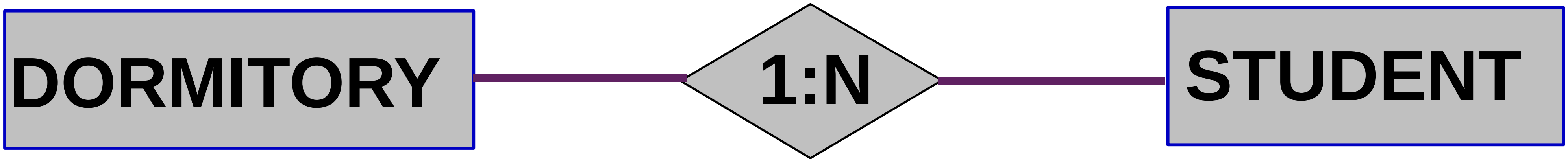


EMPLOYEE

1:1

AUTO

AUTO-ASSIGNMENT

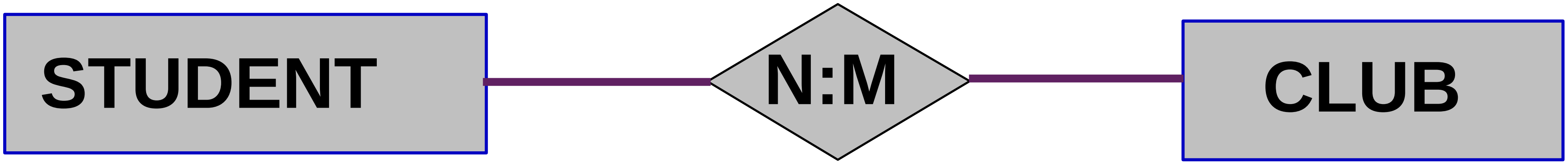


DORMITORY

1:N

STUDENT

DORM-OCCUPANT



STUDENT

N:M

CLUB

STUDENT-CLUB

# The 1:1 AUTO-ASSIGNMENT (BINARY) RELATIONSHIP

- No employee has more than one auto assigned to him/her.
- No automobile is assigned to more than one employee.

# The 1:N DORM OCCUPANT RELATIONSHIP

- A Dormitory will have "many" student occupants (where "many" could be any number: 0, 1, 5, 72, etc.).

- A Student will be an occupant of (at most) ONE dormitory.

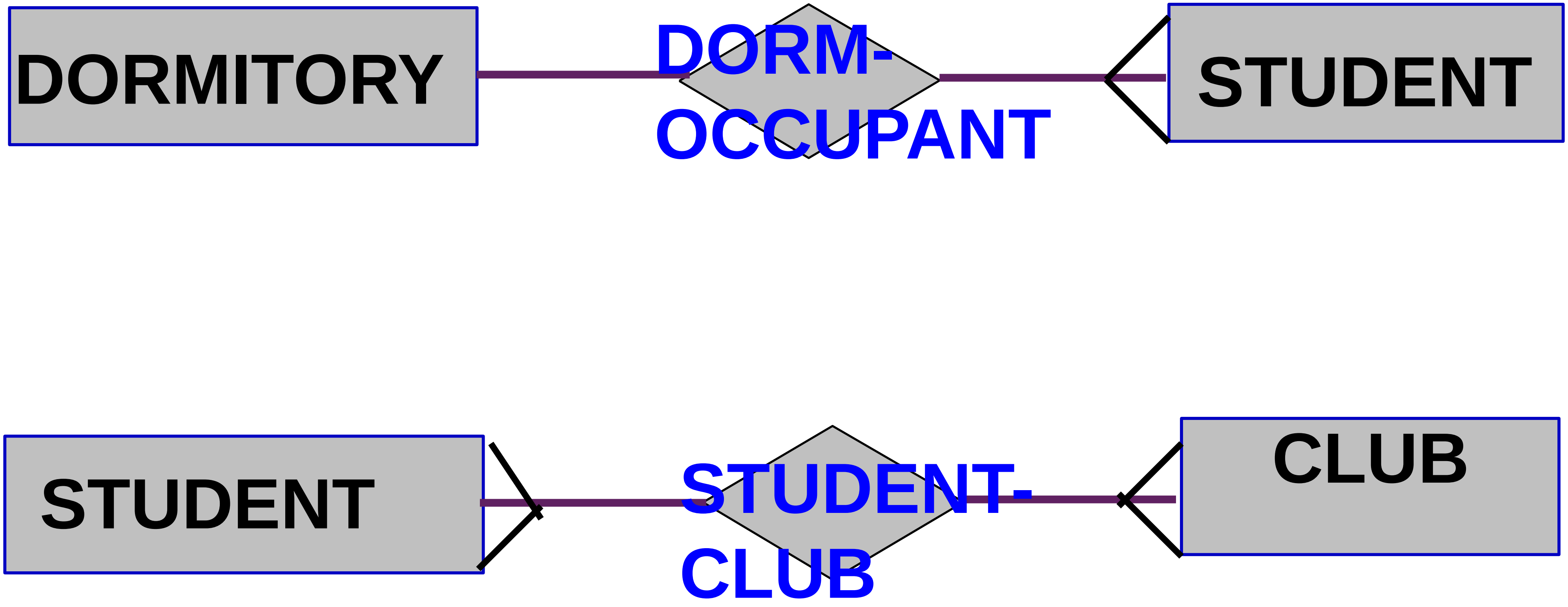
- The positions of the 1 and the N are significant. The 1 refers to the dormitory side of the relationship, and the N refers to the student side of the relationship.

# The N:M STUDENT-CLUB RELATIONSHIP

- A student can be a member of "many" (any number of) clubs.

- A club will have "many" (any number of) student members.

# An Alternative ("Crow's Foot) Notation for Denoting the "Many" Part of the Relationship



The "many" relationship cardinality is denoted by placing crow's feet (two short diagonal lines) on the many side of the relationship.

# The Maximum Cardinality of a Relationship

- **RULE:** The numbers inside the relationship diamond show the maximum number of entities which can occur on one side of the relationship.

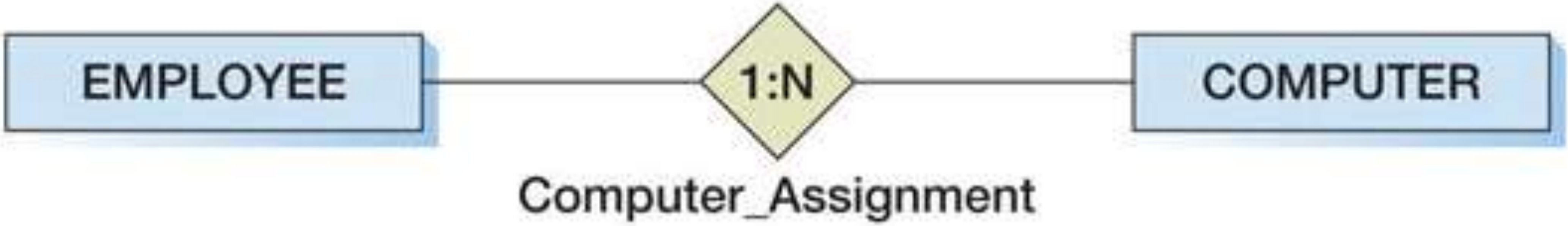
- These relationships are often called "HAS A" relationships (because, for instance, a student 'has a' dormitory). We will contrast this later with "IS A" relationships.

# Cardinality

- **Cardinality** means “count,” and is expressed as a number.
- **Maximum cardinality** is the maximum number of entity instances that **can** participate in a relationship.
- **Minimum cardinality** is the minimum number of entity instances that **must** participate in a relationship

# Parent and Child Entities

- In a one-to-many relationship:
  - The entity on the one side of the relationship is called the **parent entity** or just the **parent**.
  - The entity on the many side of the relationship is called the **child entity** or just the **child**.
- In the figure below, EMPLOYEE is the parent and COMPUTER is the child:



# HAS-A Relationships

- The relationships we have been discussing are known as **HAS-A relationships**:
  - – Each entity instance *has a* relationship with another entity instance:
    - An EMPLOYEE *has one or more* COMPUTERs.
    - A COMPUTER *has an* assigned EMPLOYEE.

# Describing the Minimum Number of Entities to Participate in a Relationship

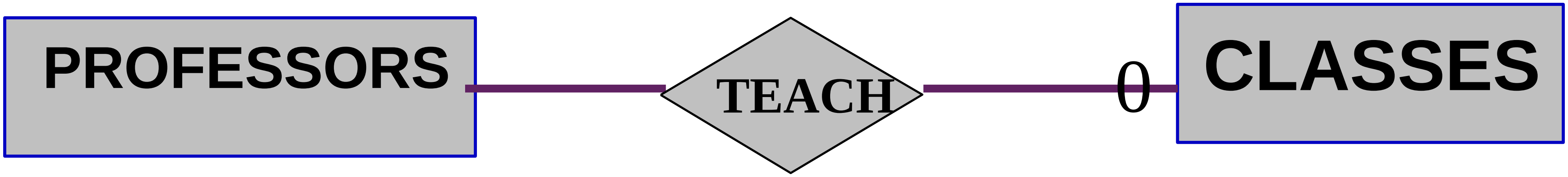
- The maximum cardinality denotes the maximum number of entities that can be involved in a relationship.
  - ***As described previously, this is usually denoted by the two values inside the diamond, separated by a colon, such as 1:N***
- We need a methodology to also describe the minimum cardinality in a relationship
  - ***For instance, should we require that a student be assigned to some dormitory, or should we make this optional for a student (we have to model the rules of the college!)***

# Relationship Participation: Case 1 - Optional Participation

- **A participating entity in a relationship is either OPTIONAL or MANDATORY**
  - **Participation is OPTIONAL if one entity occurrence does not REQUIRE a corresponding entity occurrence from the other entity set, in a particular relationship.**

# An Example of Optional Participation

- Example: At Marvel College, research professors are employed who work only on research projects and do no teaching, so these professors do not teach any classes.
  - We show an OPTIONAL entity by drawing a small circle on the side of the entity set which is to be optional.

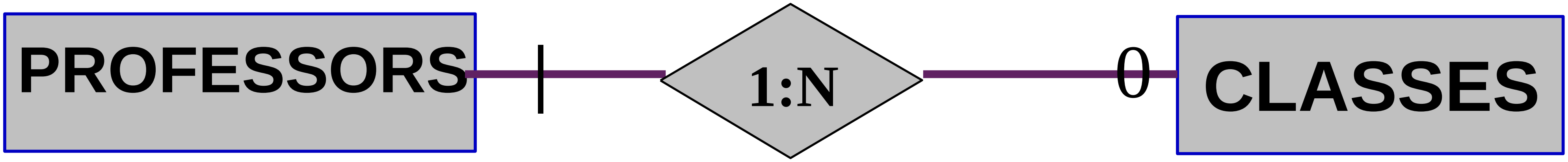


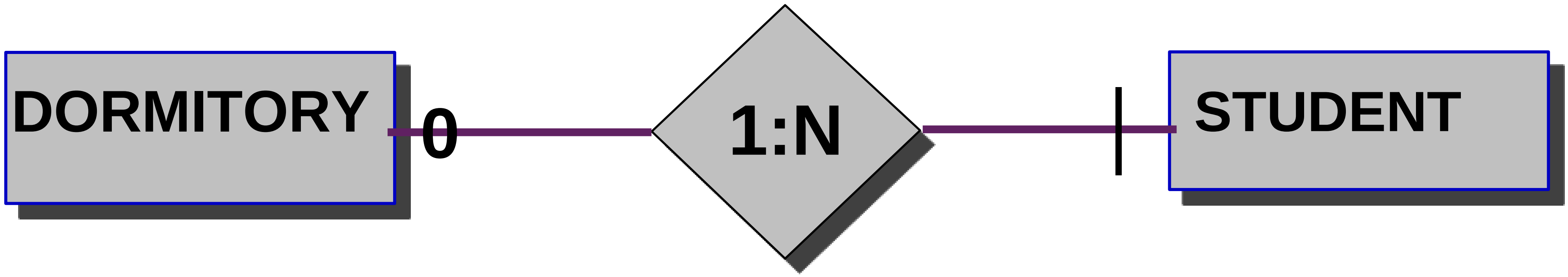
# Relationship Participation: Case II  - Mandatory Participation

- **Participation is MANDATORY in a relationship if one entity occurrence DOES require a corresponding entity occurrence in the other entity set, in a particular relationship. A mandatory participation implies a (non-zero) minimum cardinality.**

# Example of Mandatory Participation

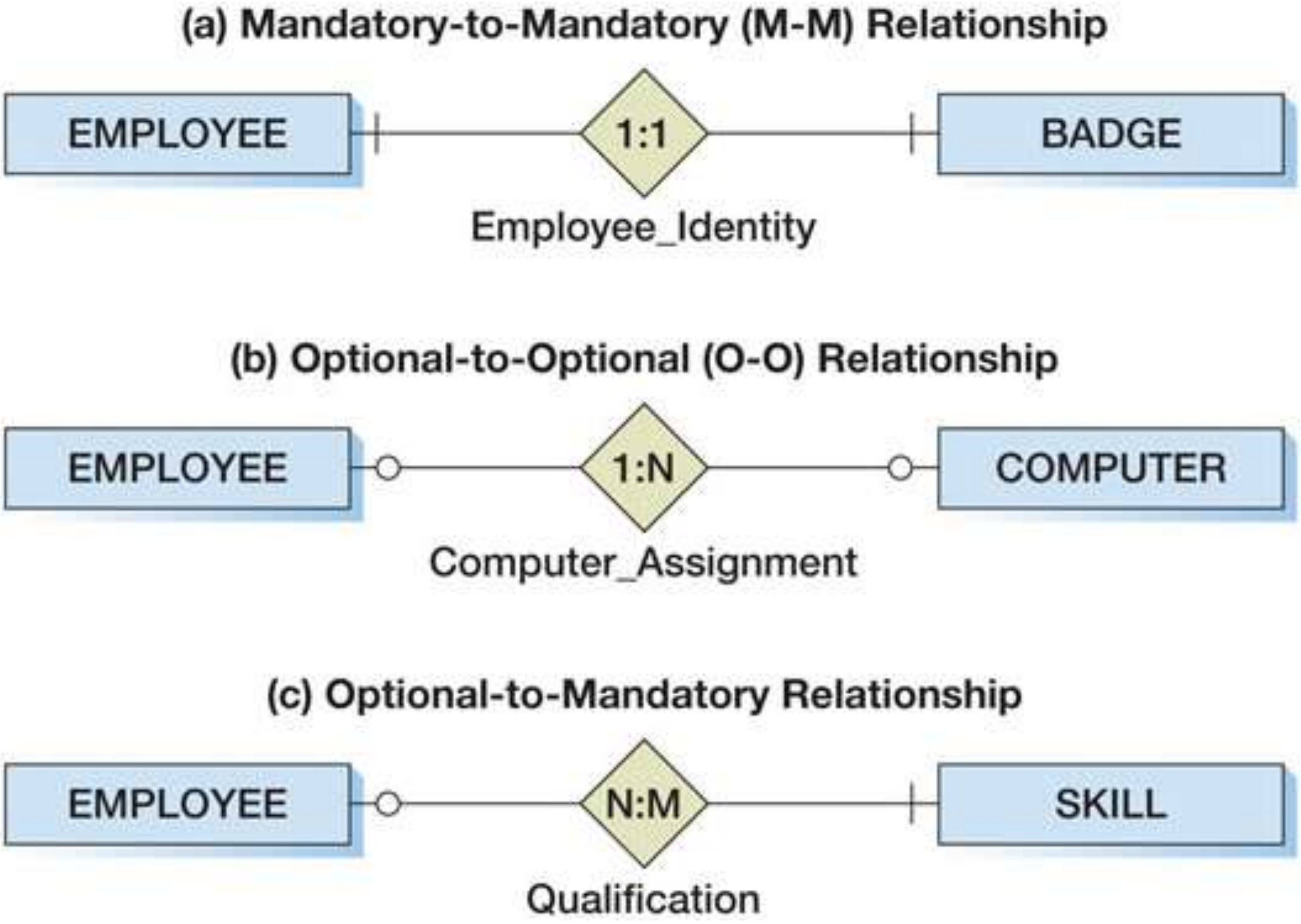
- We specify, in our E-R diagram, a mandatory participation by placing a small (usually vertical) line across the relationship line.
  - Example - Marvel College requires that each class be taught by one, and only one, professor





# The DORM-OCCUPANT Relationship with Minimum Cardinality Shown

# The Three Types of Minimum Cardinality

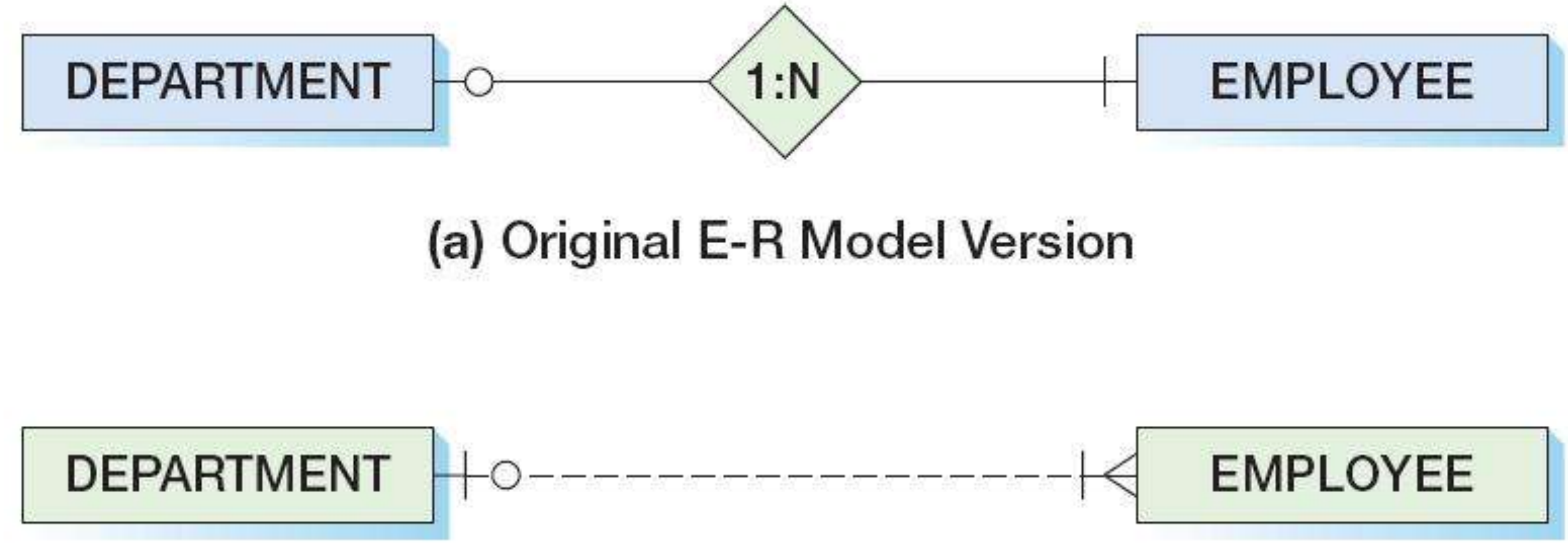


(a) Mandatory-to-Mandatory (M-M) Relationship

(b) Optional-to-Optional (O-O) Relationship

(c) Optional-to-Mandatory Relationship

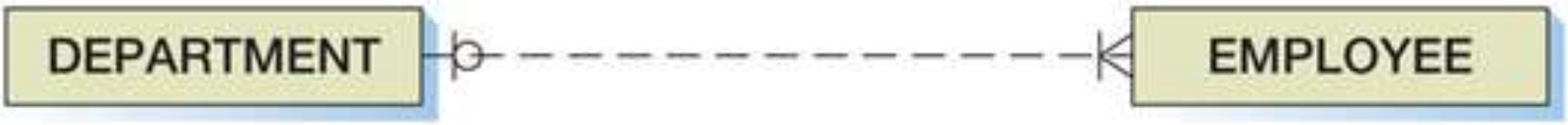
# E-R Data Modeling and the Crow's Foot Notation



(a) Original E-R Model Version

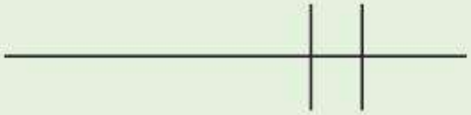
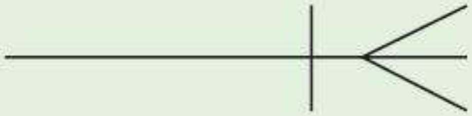
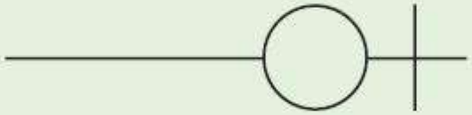
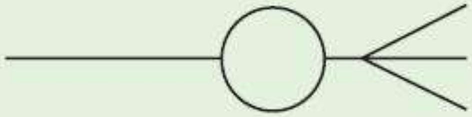
(b) Crow's Foot Version

# Data Modeling With The Crow's Foot Notation

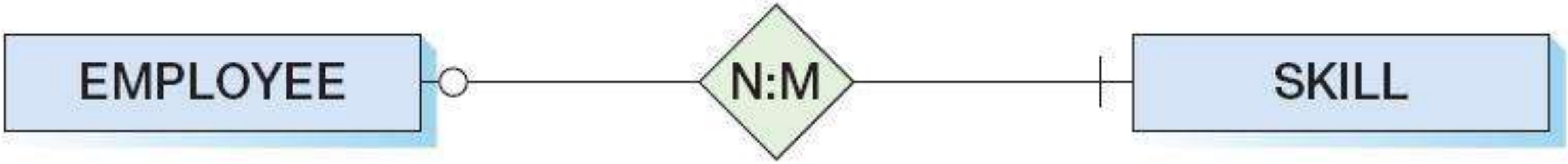


A DEPARTMENT must have at least one, and possibly many employees, while an EMPLOYEE may be associated with no DEPARTMENT or possibly with ONE Department.

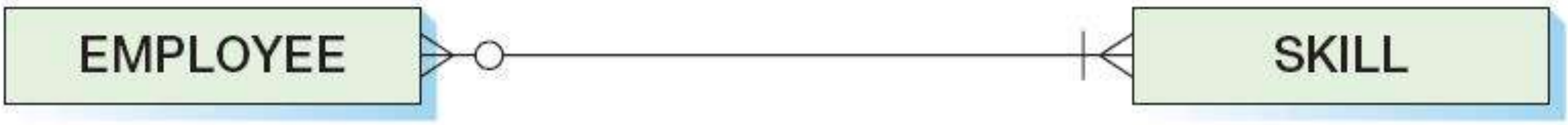
# Crow’s Foot Notation

| Symbol | Meaning |
|---|---|
|  | One—Mandatory |
|  | Many—Mandatory |
|  | One—Optional |
|  | Many—Optional |

# A Many to Many Relationship in the Original E-R model, and the Crow's Foot Version



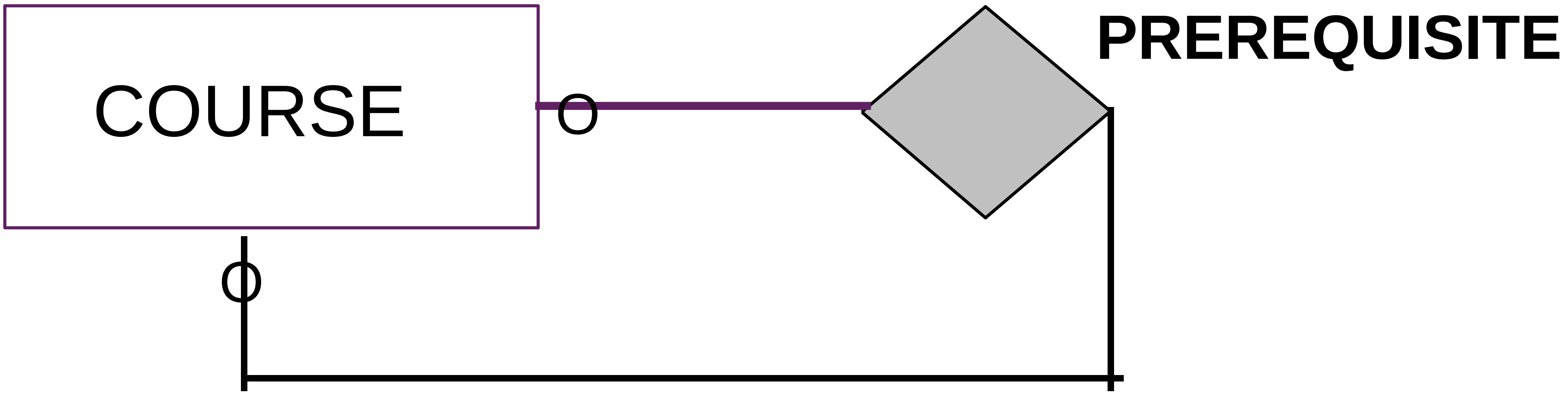
(a) Original E-R Model Version



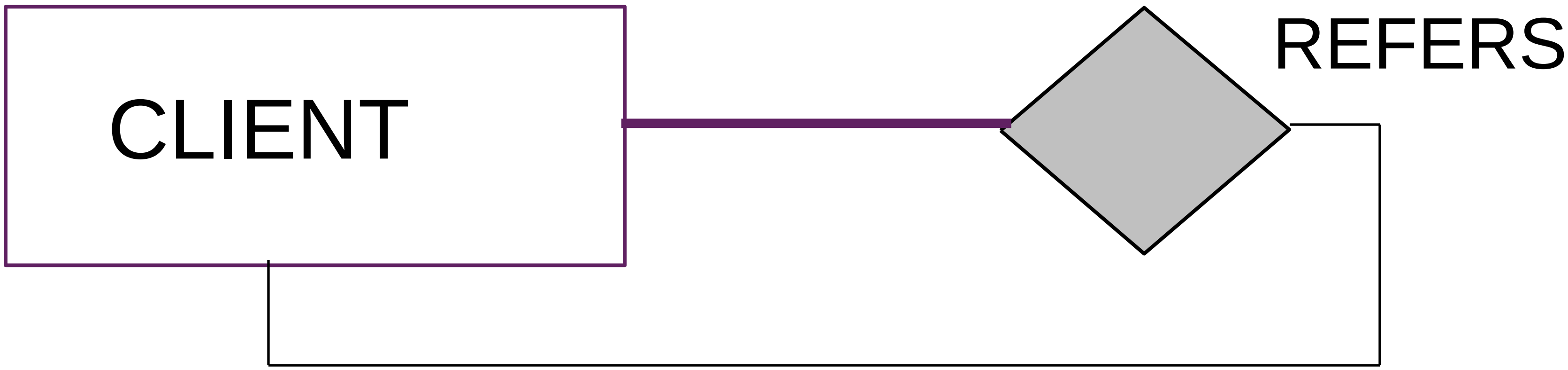
(b) Crow's Foot Version

# Cardinalities in a Recursive Relationship

One course may (or may not) serve as a prerequisite to other courses, and a given course may or may not have prerequisite courses. This is why we have the “o” in the E-R statement below.

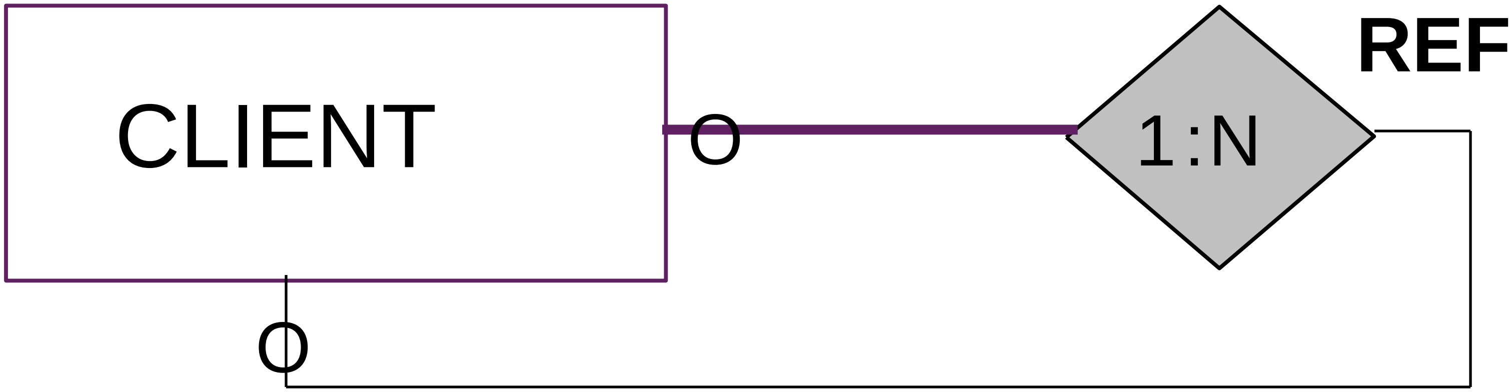


# Cardinalities in a Recursive Relationship



**Suppose the database is designed so that for each new client, we ask for the name of the (satisfied) client who referred him/her (and only one name). Some clients may not have been referred by other clients.**

# Cardinalities in a Recursive Relationship



**Suppose the database is designed so that for each new client, we ask for the name of the (satisfied) client who referred him/her (and only one name). Some clients may not have been referred by other clients.**

# Displaying Attributes in E-R Diagrams

- **Usual way: attributes appear in ellipses and are connected to the entity set or relationship to which they belong**

- **Alternative way (when there are a significant number of attributes for the entity sets) is to list them separately beneath the E-R Diagram, or to include them in the portrait rectangle representing the Entity Set**

# Strong Entities

- A strong entity (or entity set) is an entity that can exist on its own, and its existence does not depend on the presence of another entity.
  - PERSON and AUTOMOBILE are examples of strong entities

- This is in contrast to weak entities.

# Weak Entities

- **A WEAK ENTITY is an entity whose existence in the database depends upon the existence of some other entity.**
  - ➔ ***Example - Insurance companies provide medical coverage to the teachers at Sleepy Valley High and their dependents. So, a dependent can exist in the database only if he/she is associated with a teacher at Sleepy Valley High School.***

- **A weak entity set is denoted in an E-R model by rounding the corners of the entity rectangle.**

# ID-Dependent Entities

- An **ID-dependent entity** is an entity (child) whose identifier includes the identifier of another entity (parent).

- Consider APARTMENTs in a BUILDING. Each building will have an identifier (BuildingAddress), but if apartments have numbers, like 4E,5B, etc., the apartment number is NOT a true identifier (there could be several buildings having the same apartment number, but the composite (BuildingAddress, ApartmentNumber) will serve as an identifier. Apartment is thus an ID-Dependent entity, because its identifier includes the identifier of another entity (its parent).

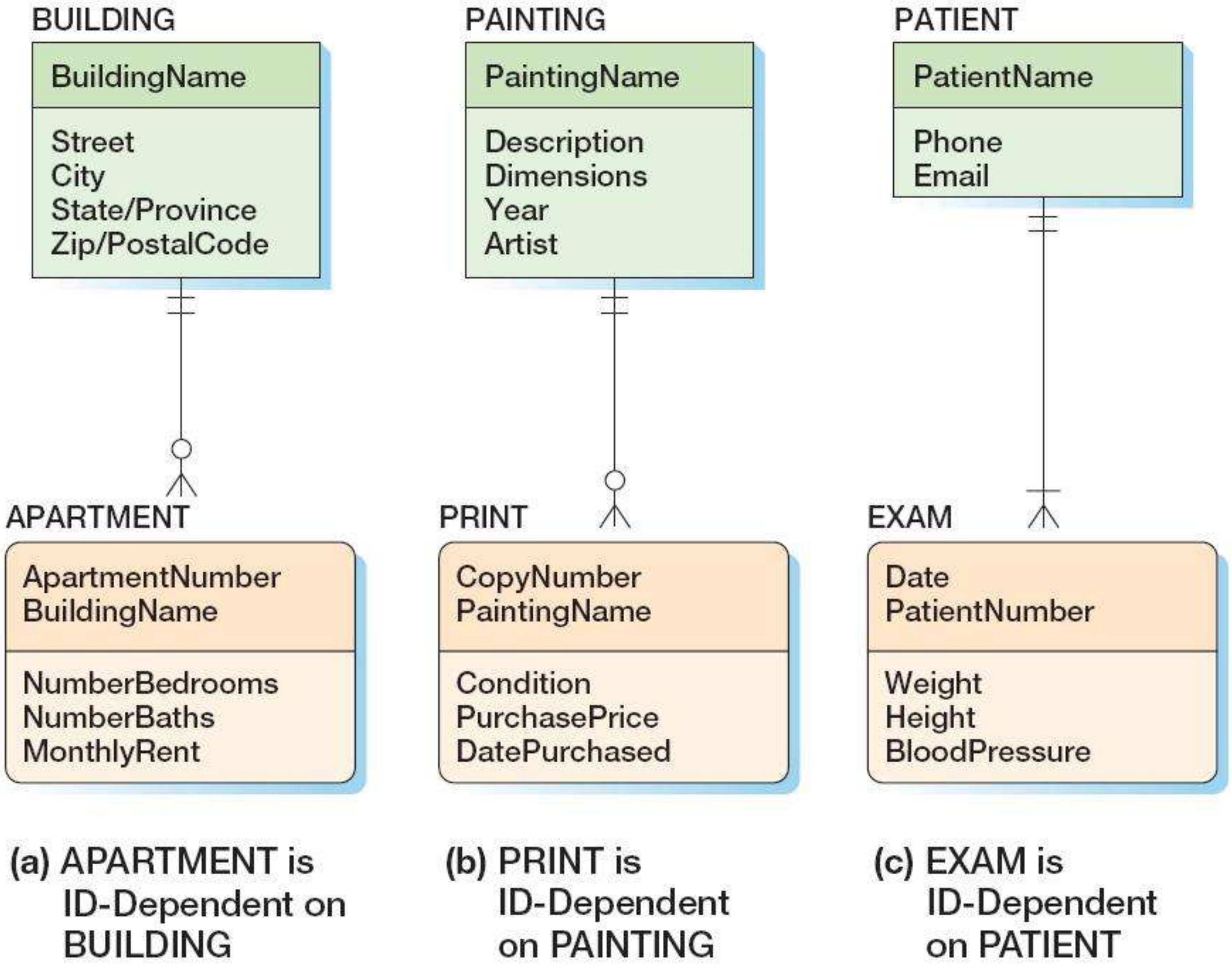
- The minimum cardinality from the ID-dependent entity to the parent is always one.

- All ID-Dependent entities are considered weak, and are thus drawn as rectangles with rounded corners.

# ID-dependent Entities

- The identifier of APARTMENT would be the composite identifier (BuildingAddress, ApartmentNumber).
  - **So, both logically and physically, an APARTMENT cannot exist unless a BUILDING exists**.
- Another example of an ID-dependent entity is the entity VERSION in the relationship between PRODUCT and VERSION, where PRODUCT is a software product, and VERSION is a release of that software product.
  - **Identifier of PRODUCT would be ProductName, and identifier of the ID-dependent entity set VERSION would be (Productname, ReleaseNumber)**

# ID-Dependent Entities



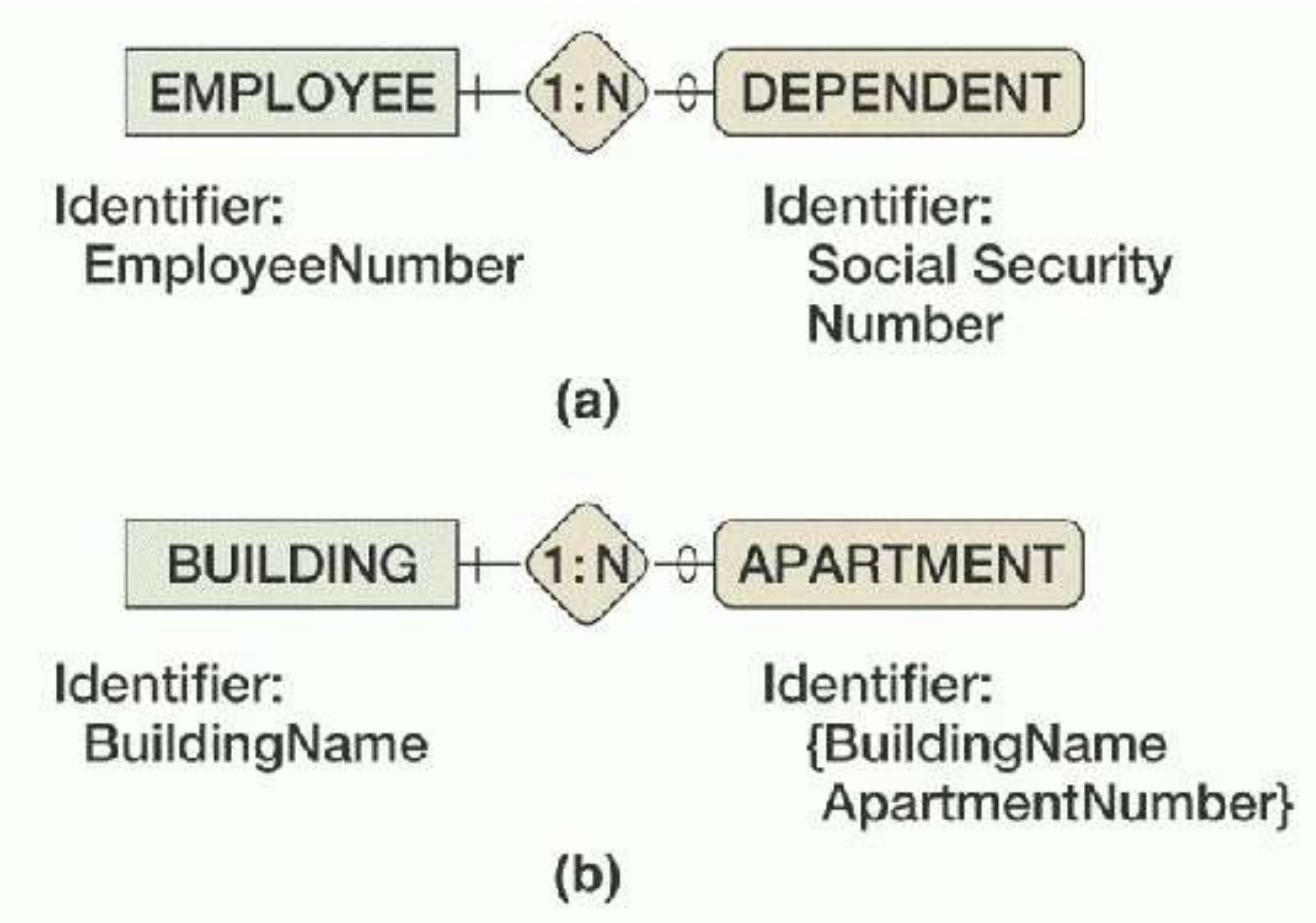
(a) APARTMENT is ID-Dependent on BUILDING

(b) PRINT is ID-Dependent on PAINTING

(c) EXAM is ID-Dependent on PATIENT

Note that a BUILDING may exist without an APARTMENT, and a PAINTING may exist without a PRINT, but, as per this data model, a PATIENT may not exist in the database without an associated EXAM.

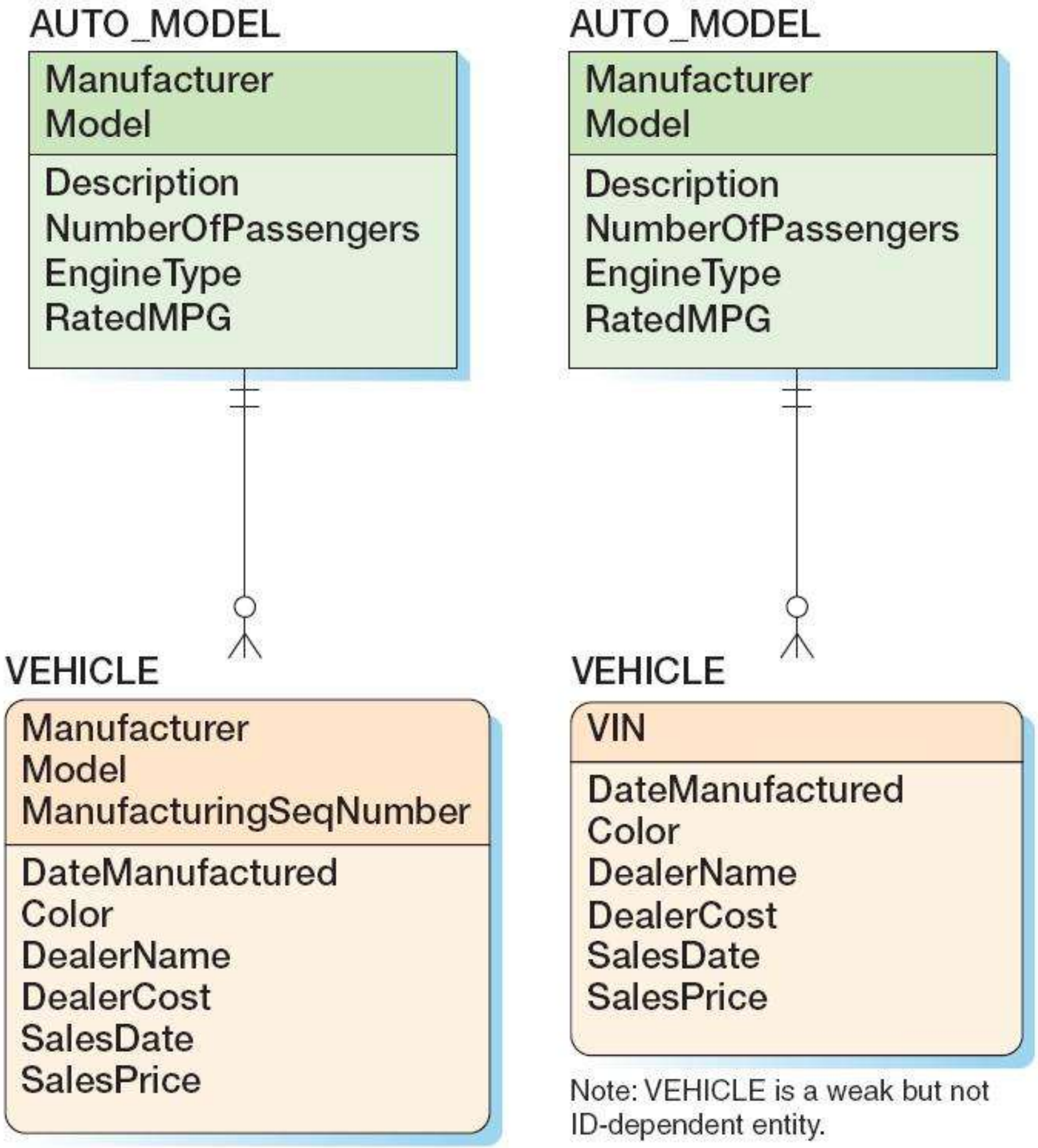
# Weak Entities (Summary)



**Weak, but not ID-dependent**

**Weak and ID-dependent**

# Weak Entities Examples



AUTO_MODEL

Manufacturer
Model

Description
NumberOfPassengers
EngineType
RatedMPG

VEHICLE

Manufacturer
Model
ManufacturingSeqNumber

DateManufactured
Color
DealerName
DealerCost
SalesDate
SalesPrice

(a) ID-Dependent Entity

AUTO_MODEL

Manufacturer
Model

Description
NumberOfPassengers
EngineType
RatedMPG

VEHICLE

VIN

DateManufactured
Color
DealerName
DealerCost
SalesDate
SalesPrice

Note: VEHICLE is a weak but not ID-dependent entity.

(b) Non-ID-Dependent Weak Entity

# Explanation of the Auto-Model Example

- In (a), each vehicle is assigned a sequential number as it is manufactured. So, we have a Toyota Camry 1, a Toyota Camry 2, etc. (where Toyota is the manufacturer, and Camry is the model). So, VEHICLE is an ID-Dependent Entity, because it contains the identifier (Manufacturer, Model) of the entity set AUTO_MODEL.
- But, in (b), the VIN is an identifier, so VEHICLE is not an ID-Dependent entity. But, VEHICLE is weak, because if the auto-model Camry did not exist, then the VEHICLE would not exist.

# Some Salient Points Regarding Weak Entities

- To be a weak entity, an entity must ***logically*** depend on another entity.
  - *Not all entities that have a minimum cardinality of 1 in a relationship are weak – only those that are logically dependent are weak*
- All ID-dependent entities are weak.
- Every weak entity has a minimum cardinality of 1 on the entity upon which it depends.
- An entity with a minimum cardinality of 1 need not necessarily be weak.

# Subtypes and Supertypes

- **In some data modeling applications, an entity set may contain optional sets of attributes.**
  - **Example: Suppose the entity set CLIENT has the following required attributes:**
    - **Client Number**
    - **Client Name**
    - **Amount Due**
  - **Suppose also that there are three different categories of CLIENT:**
    - **Individual-Client**
    - **Partnership Client**
    - **Corporate-Client**

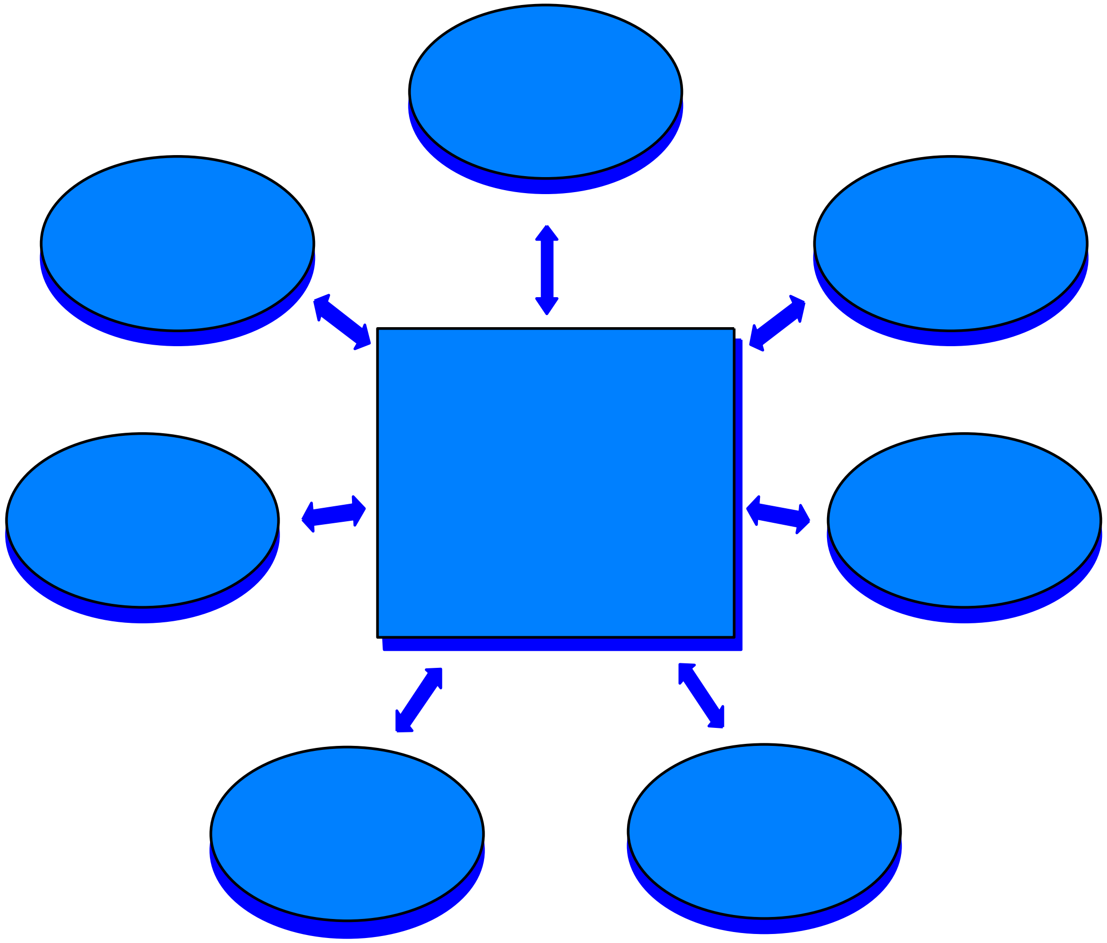
# Additional Attributes for Specific Client Categories

- **Individual-Client entities require the following additional attributes:**
  - ➔ Address
  - ➔ Social Security Number
- **Partnership-Client entities require these additional attribures:**
  - ➔ ManagingPartnerName
  - ➔ Address
  - ➔ TaxIdentificationNumber
- **Corporate-Client entities require these additional attributes:**
  - ➔ Contact Person
  - ➔ Phone
  - ➔ TaxIdentificationNumber

# One (<u>not-recommended</u>) Approach for Modeling Client in an E-R Diagram

If we allocate ALL attributes to the entity set CLIENT, then instances of CLIENT would leave some attributes blank. This would generate a signficant number of nulls for attribute values. The major reason for creating subtypes is to avoid these value-inappropriate nulls.

Example: If Social-Security number were one of the attributes, then every corporate and partnership client would have a null value for social-security number.



# E-R Modeling of CLIENT using Subtypes and Supertypes

- The Extended E-R Model introduced the concepts of supertypes and subtypes.
- Make CLIENT a supertype, with attributes ClientNumber, ClientName and AmountDue.
- Make INDIVIDUAL-CLIENT, PARTNERSHIP-CLIENT and CORPORATE-CLIENT subtypes of the supertype CLIENT, each having their own dedicated set of attributes, but “inheriting” the common attributes from the supertype/parent CLIENT.

# Inheritance

- **Inheritance means that the entities in the subtypes "inherit" the attributes of the supertype entity class.**

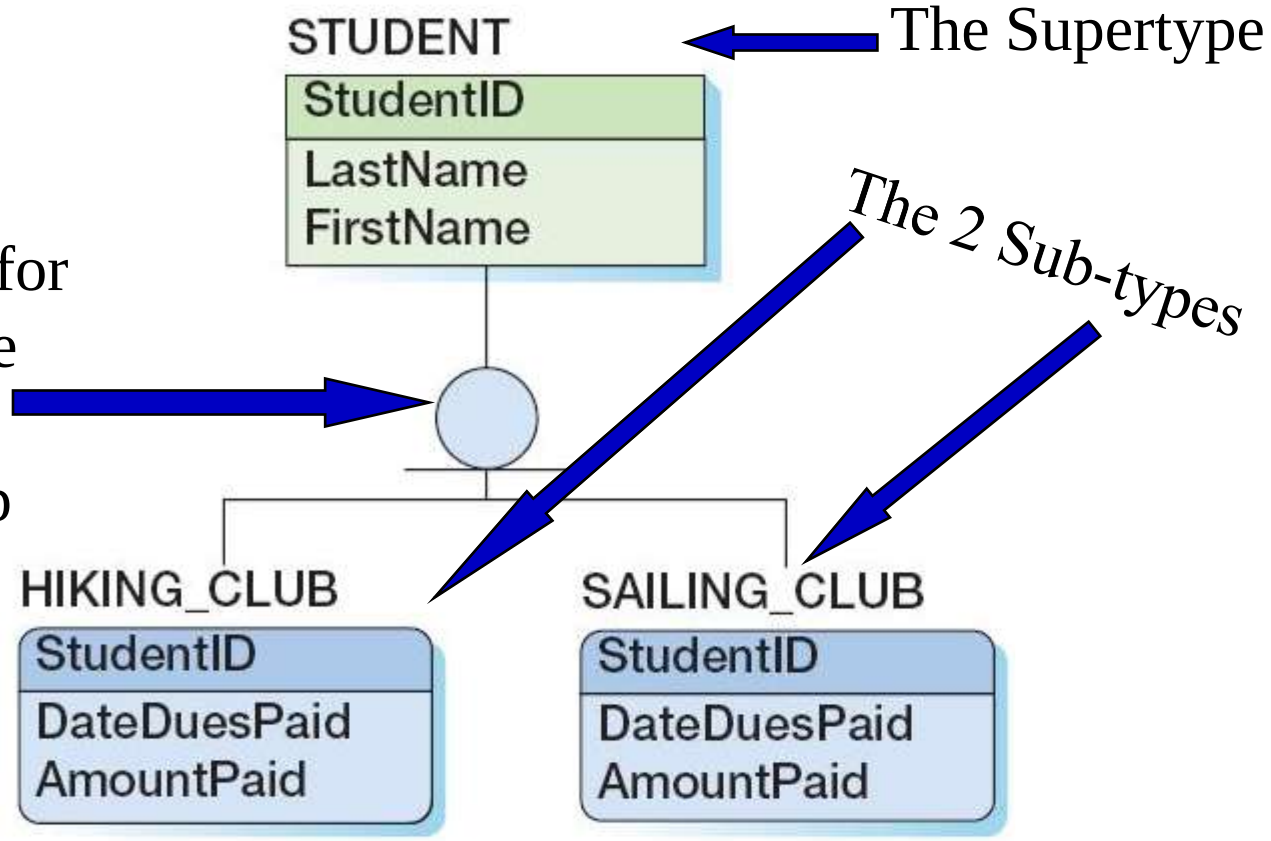
  - **Thus, CORPORATE-CLIENT inherits the attributes Client-Number,Client-Name and AmountDue from CLIENT.**

# Subtype Entities

- A **subtype entity** is a child, having a parent **supertype entity from which it inherits attributes**:
  - Example: STUDENTs can be classified as either UNDERGRADUATE or GRADUATE. We can make STUDENT a supertype and it will be the “parent” to the subtypes UNDERGRADUATE and GRADUATE.
- The supertype STUDENT contains all common attributes for the subtypes UNDERGRADUATE and GRADUATE, though the UNDERGRADUATE and GRADUATE subtypes may also have their own dedicated attributes.
- A **discriminator** is an attribute that determines which subtype is appropriate.
  - Example : A discriminator called GENDER could determine whether a MALE subtype or FEMALE subtype is apropos.

# Denoting Supertypes and Subtypes in Our ER Diagrams

- We place the supertype entity set above the subtype entity sets, and use a circle with a line beneath it to denote a supertype-subtype relationship. We use a solid line to represent an ID-dependent subtype entity, since each subtype is ID-dependent on the supertype.

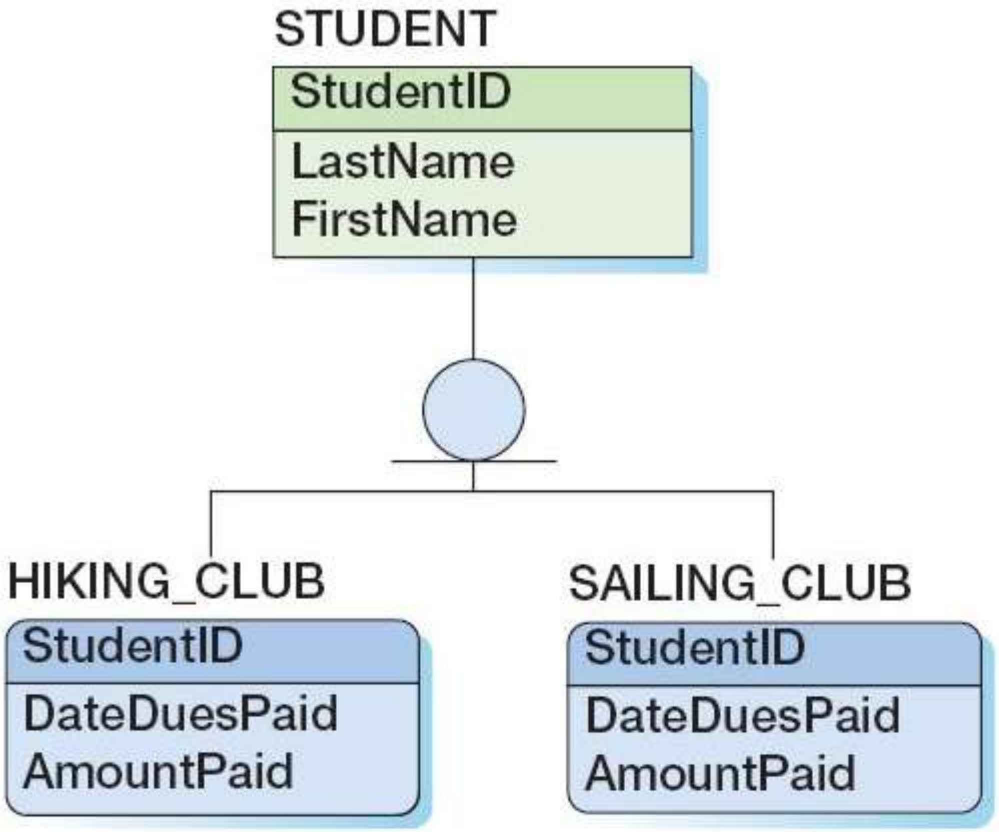


The Supertype

The 2 Sub-types

The notation for a super-type subtype relationship

# Explaining the Hiking Club and the Sailing Club



The HIKING_CLUB subtype consists of student members, and while the subtype has attributes for the date the dues were paid and the amount paid, it "inherits" from its parent STUDENT the last name and first name of its members.

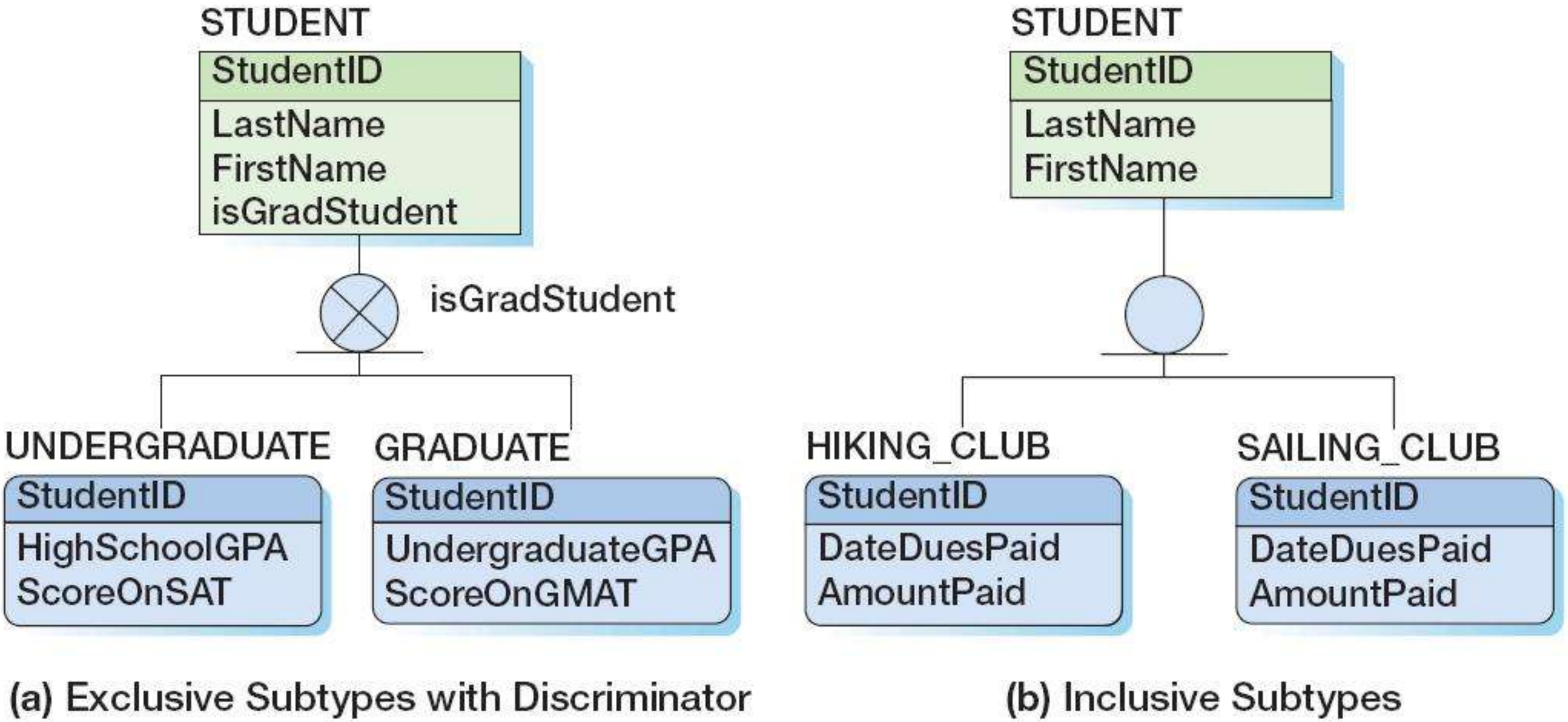
The same is true for student members of the SAILING-CLUB.

The notation states that a given student may either be in the hiking club, or the sailing club, or both (or neither).

# Example Subtype Entity Sets

In (a) below, a STUDENT is either an UNDERGRADUATE or a GRADUATE, but not both. The “X” in the circle denotes “exclusive” (i.e., the supertype is ‘related” to at most one subtype).

In (b) below, the absence of an “X” in the circle denotes “inclusive” (i.e., the supertype can relate to any of the subtypes).



# Discriminators

- A **discriminator** is an (optional) attribute that determines which subtype is appropriate.
  - The discriminator will appear as an attribute name and will also be placed adjacent to the subtype symbol

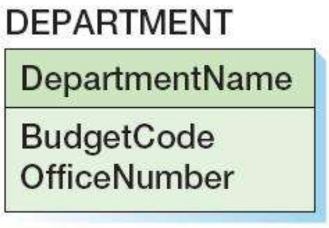
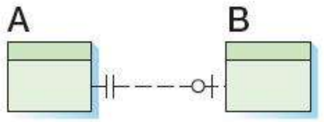
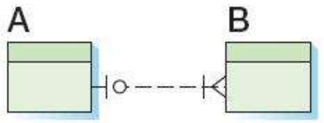
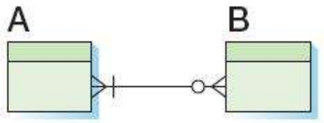
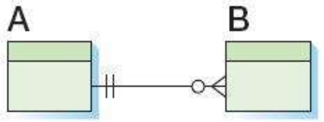
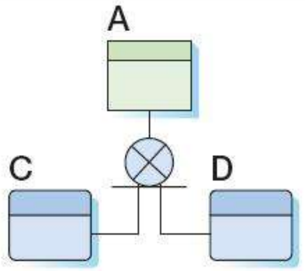
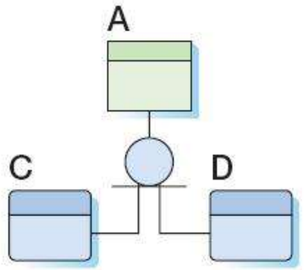
- Example: The attribute “isGradStudent,” which appears in STUDENT (diagram a) on the prior slide, **is a discriminator** and will have a domain of “Yes” and “No,” and therefore appears adjacent to the subtype symbol.

# Subtypes: IS-A relationships

- Relationships connecting supertypes and subtypes are called **IS-A relationships**, because a subtype IS really the same entity as the supertype.
  - Contrast this with a "HAS-A" relationship, where we have a relationship involving two entity sets, but the identifiers of the entity sets are different.

- The identifer of the supertype and all of its subtypes must be identical, i.e., the identifier of the supertype becomes the identifier of the related subtype(s).

# Crow’s Foot Nomenclature Summary

| | |
|---|---|
| <br> | DEPARTMENT entity; DepartmentName is identifier; BudgetCode and OfficeNumber are attributes. |
| <br> | 1:1, nonidentifying relationship. A relates to zero or one B; B relates to exactly one A. Identifier and attributes not shown. |
| <br> | 1:N, nonidentifying relationship. A relates to one or many Bs; B relates to zero or one A. Identifier and attributes not shown. |
| <br> | Many-to-many, nonidentifying relationship. A relates to zero or more Bs; B relates to one or more As. |
| <br> | 1:N identifying relationship. A relates to zero, one, or many Bs. B relates to exactly one A. Identifier and attributes not shown. For identifying relationships, the child must always relate to exactly one parent. The parent may relate to zero, one, many, or a combination of these minimum cardinalities. |
| <br> | A is supertype, C and D are exclusive subtypes. Discriminator not shown. Identifier and attributes not shown. |
| <br> | A is supertype, C and D are inclusive subtypes. Identifier and attributes not shown. |

# An Example of the Data Modeling Process

- The Database Development team, when they commence the data modeling process,
  - will analyze user requirements
  - will examine existing reports of the organization
  - will examine existing data entry forms of the organization
  - will examine existing data sources
  - will conduct interviews

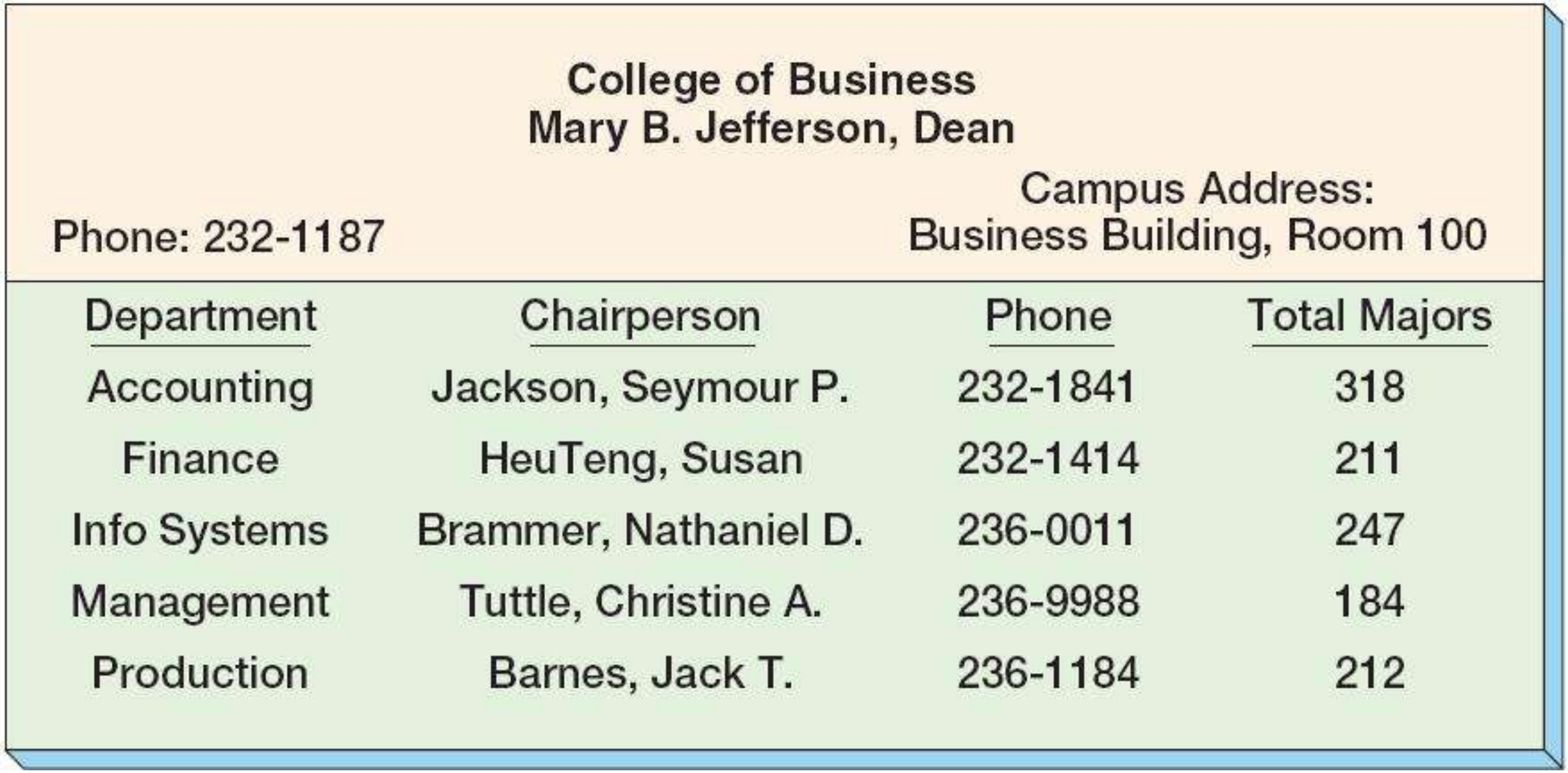
- As part of the process, users may be asked for additional information (e.g., to determine cardinalities…)

- Users will review and validate the prototype data model

- The process will be illustrated by a case study of Highline University, which is seeking to create a database to track its colleges, departments, faculty and students.

# The College Report

- The following is an existing report from the College of Business, and we are told that every college at Highline University produces a similar report.



**College of Business**<br>
**Mary B. Jefferson, Dean**

Phone: 232-1187

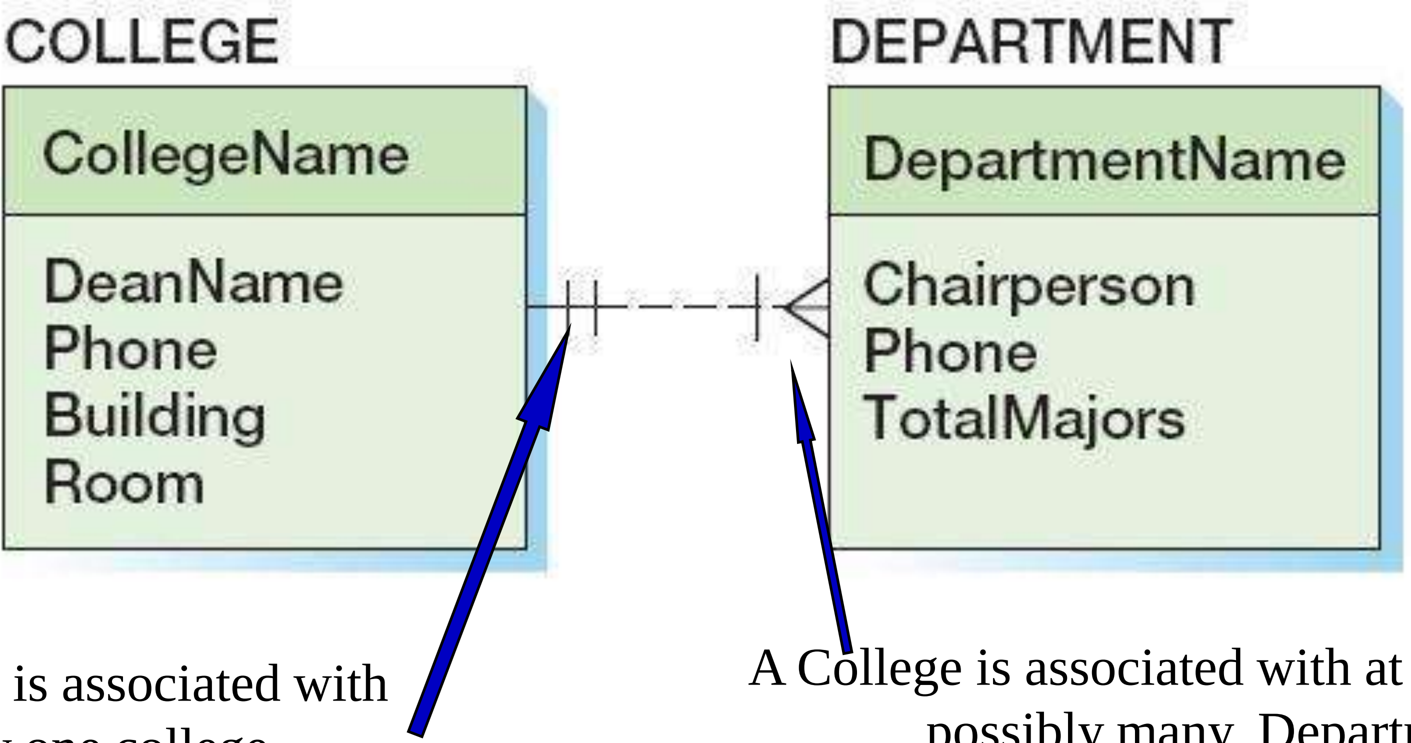
Campus Address:<br>
Business Building, Room 100

| Department | Chairperson | Phone | Total Majors |
|---|---|---|---|
| Accounting | Jackson, Seymour P. | 232-1841 | 318 |
| Finance | HeuTeng, Susan | 232-1414 | 211 |
| Info Systems | Brammer, Nathaniel D. | 236-0011 | 247 |
| Management | Tuttle, Christine A. | 236-9988 | 184 |
| Production | Barnes, Jack T. | 236-1184 | 212 |

# Examining the College Report

- The College report suggests two entity sets for our database:
  - a COLLEGE entity set, with attributes for the college name, its Dean, phone and campus address
  - a DEPARTMENT entity set, with attributes for the name of the department, its chairperson, phone, and how many majors are in the department.
  - A subsequent interview with the Highline administration indicates that a college could have one department, or several department (for instance,  the college of architecture has only one department (Architecture), while the college of engineering has several (EE, CivilE, Mechanical engineering, etc.) And, a department can only be a member of one college.
- Based upon the above information from the College Report, we create a Crow's Foot data model, as given on the following slide:

# Preliminary Data Model from the College Report



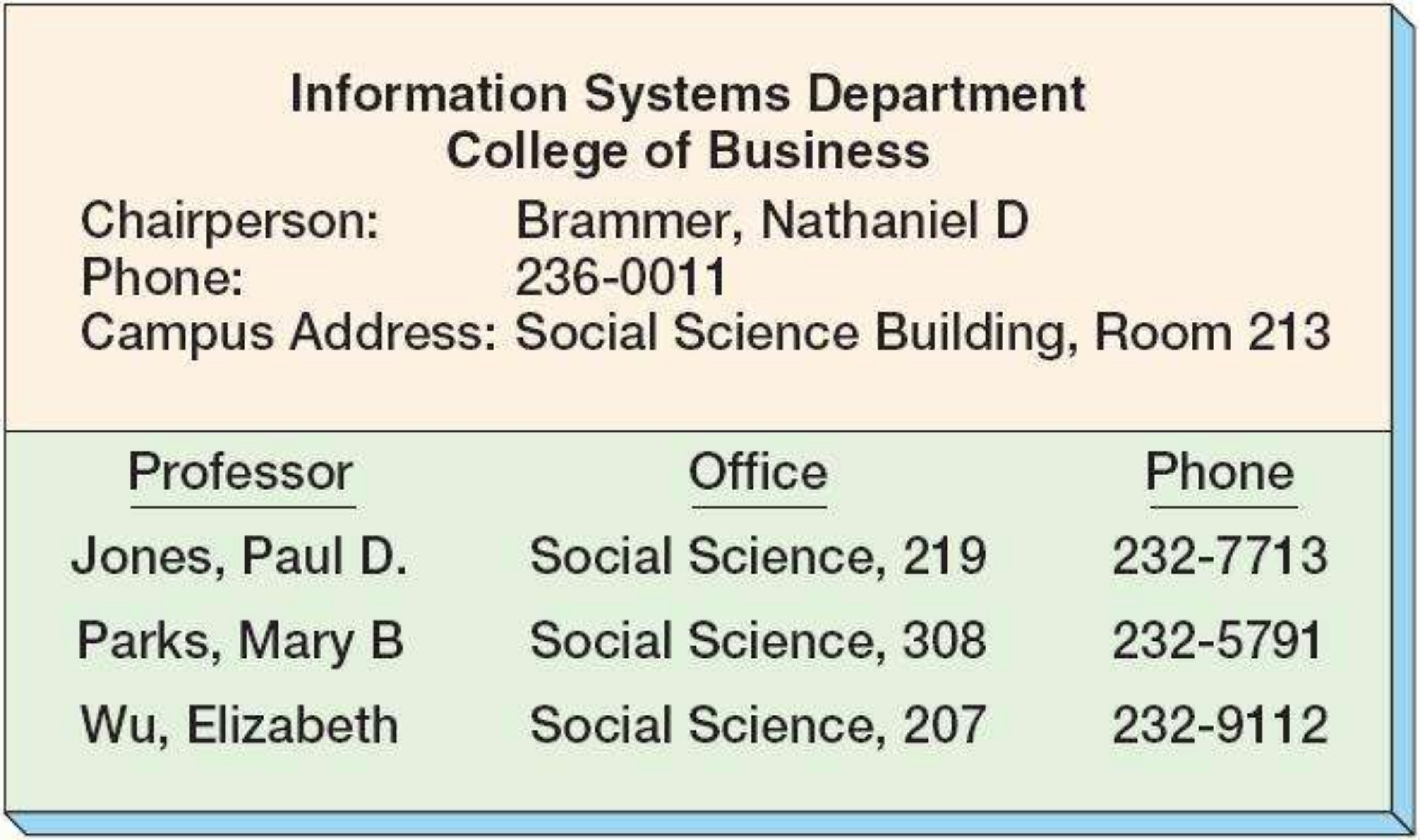
A Department is associated with precisely one college

A College is associated with at least one, and possibly many, Departments

The relationship between COLLEGE and DEPARTMENT is non-identifying, i.e., a dotted line, because DEPARTMENT is NOT ID-Dependent on College, since Department has its own identifier and not the College’s identifier.

# The Department Report

- The following is an existing report from the Department of Information Systems, and we are told that every academic department at Highline University produces a similar report.



**Information Systems Department**
**College of Business**

Chairperson: Brammer, Nathaniel D
Phone: 236-0011
Campus Address: Social Science Building, Room 213

| Professor | Office | Phone |
|---|---|---|
| Jones, Paul D. | Social Science, 219 | 232-7713 |
| Parks, Mary B | Social Science, 308 | 232-5791 |
| Wu, Elizabeth | Social Science, 207 | 232-9112 |

# Examining the Department Report

- There is a need for a PROFESSOR entity set, which would have attributes detailing the name of the professor, his her office and phone.

- Since a Professor is a member of a Department, there is also a new relationship between professors and their departments which must be added to our evolving data model.

- So, as is typical in the data modeling process, the examination of a new report generates some needed adjustments to our data model.

# Preliminary Data Model for COLLEGE-DEPARTMENT-PROFESSOR

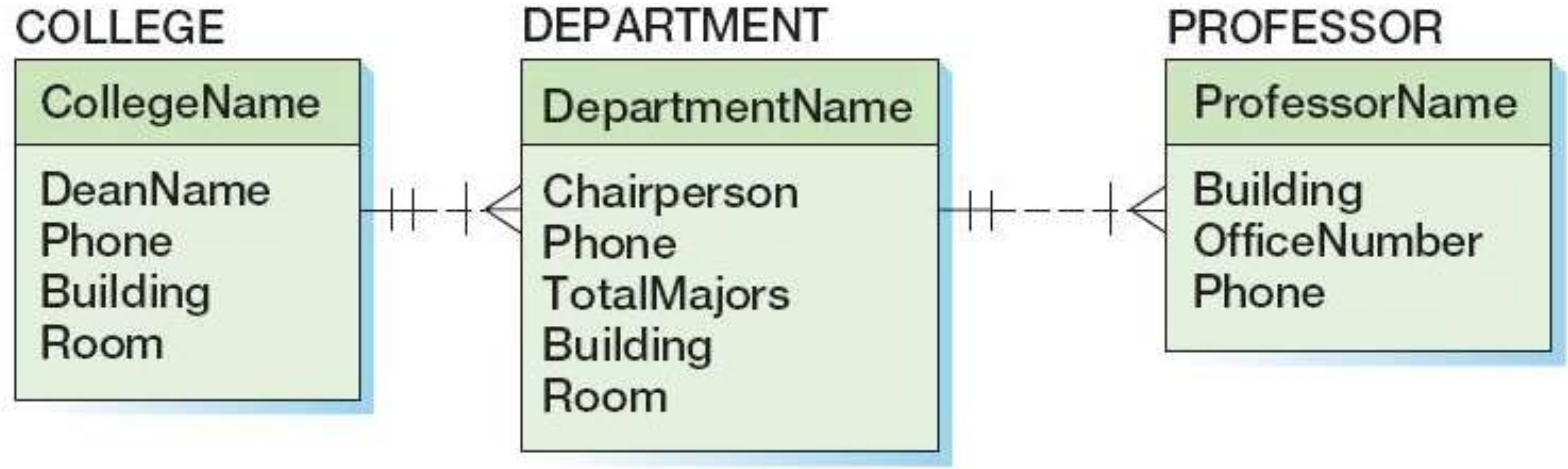
- One unanswered question for Highline University is whether or not a professor is associated with only one department, or whether a professor can teach in several departments (called a "joint appointment").

- We will develop TWO preliminary data models – one where a professor must teach in one and only one department, and a second data model where a professor may teach in two (or more) departments.

    - In the case where a professor is associated with only one department, we will have a one to many relationship between PROFESSOR and DEPARTMENT (i.e., a Department is associated with many professors, but a professor is associated with only one Department.)

    - In the case where a professor may be associated with one or more departments, we will have a many to many relationship between PROFESSOR and DEPARTMENT (i.e., a Department is associated with many professors, and a professor could be associated with many Departments.)

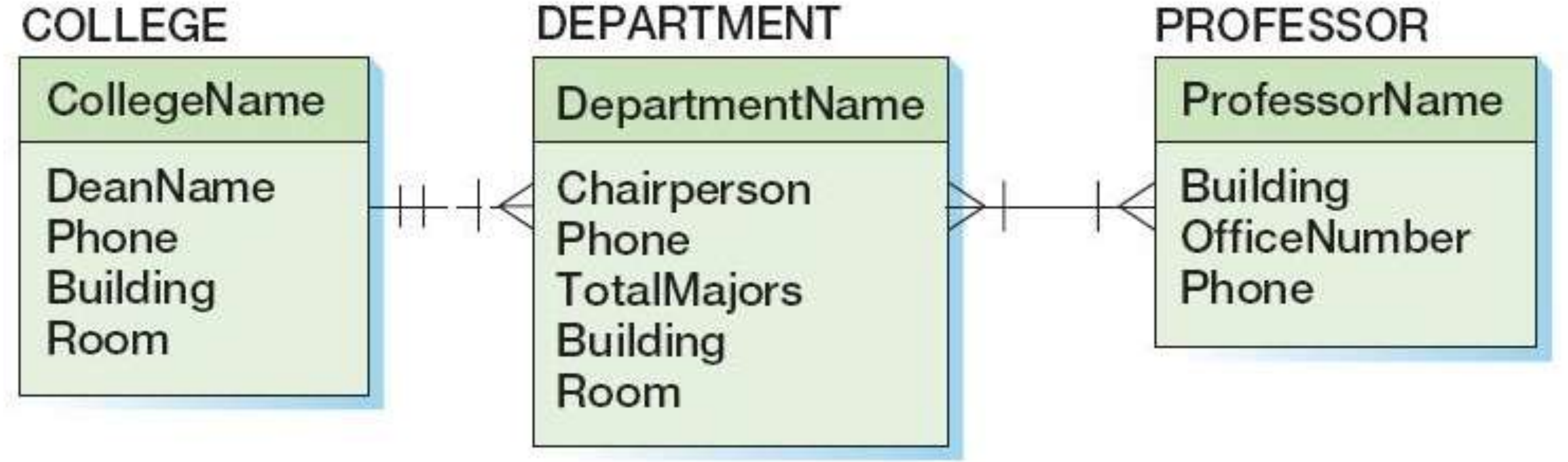
# The Evolving E-R Crow's Foot Data Model



- We have added to DEPARTMENT, based upon the Department Report, attributes which were not in the prior data model: Building and Room

- We have added the PROFESSOR entity set to the E-R data model, and for this instance we assume that a given professor is associated with exactly one department, so the Crow's Foot model has a 1 to many relationship between DEPARTMENT and PROFESSOR.

# And If A Professor May Teach In Several Departments…



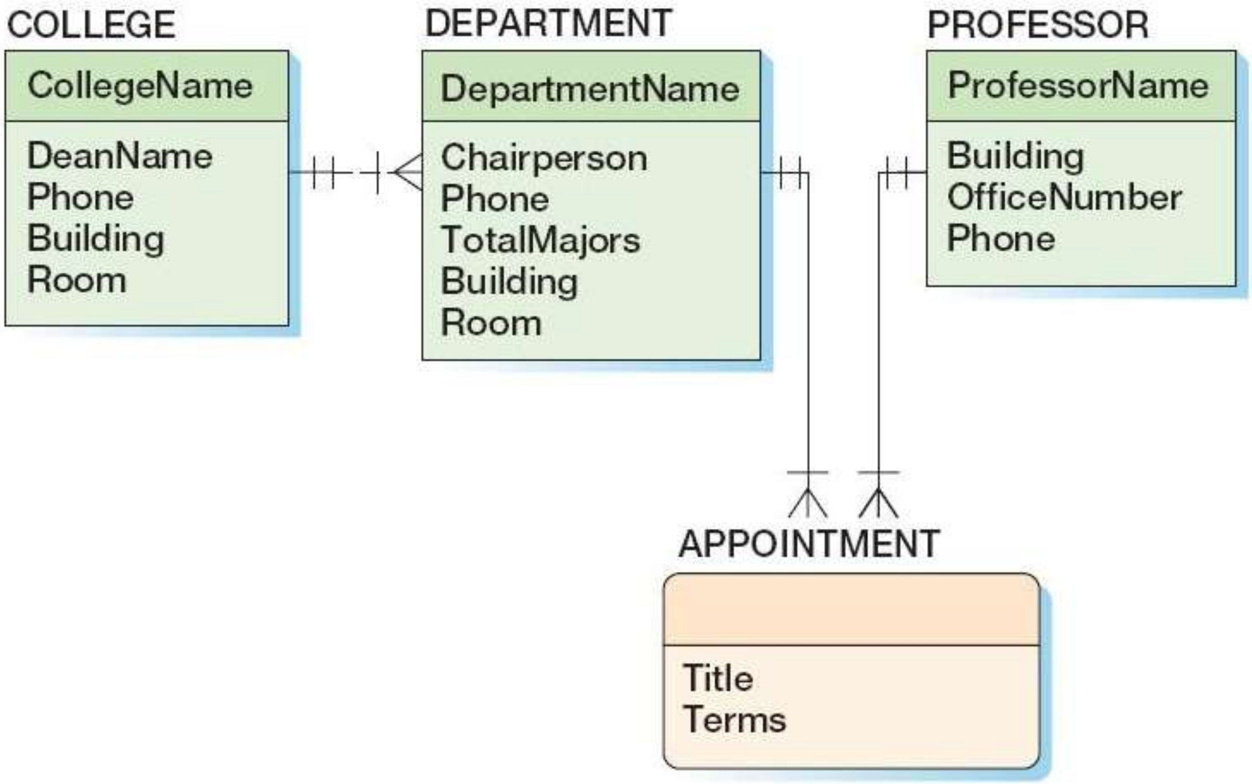
- We have made the relationship between DEPARTMENT and PROFESSOR a many-to-many relationship, by putting crow’s feet on both sides of the relationship line, and denoting required relationships on both sides (i.e., a professor must be associated with a minimum of 1 department, and a department must be associated with a minimum of 1 professor)

# Appointments of Professors to Departments

- Suppose the data modeling team uncovers files containing information regarding the appointments of professors to departments

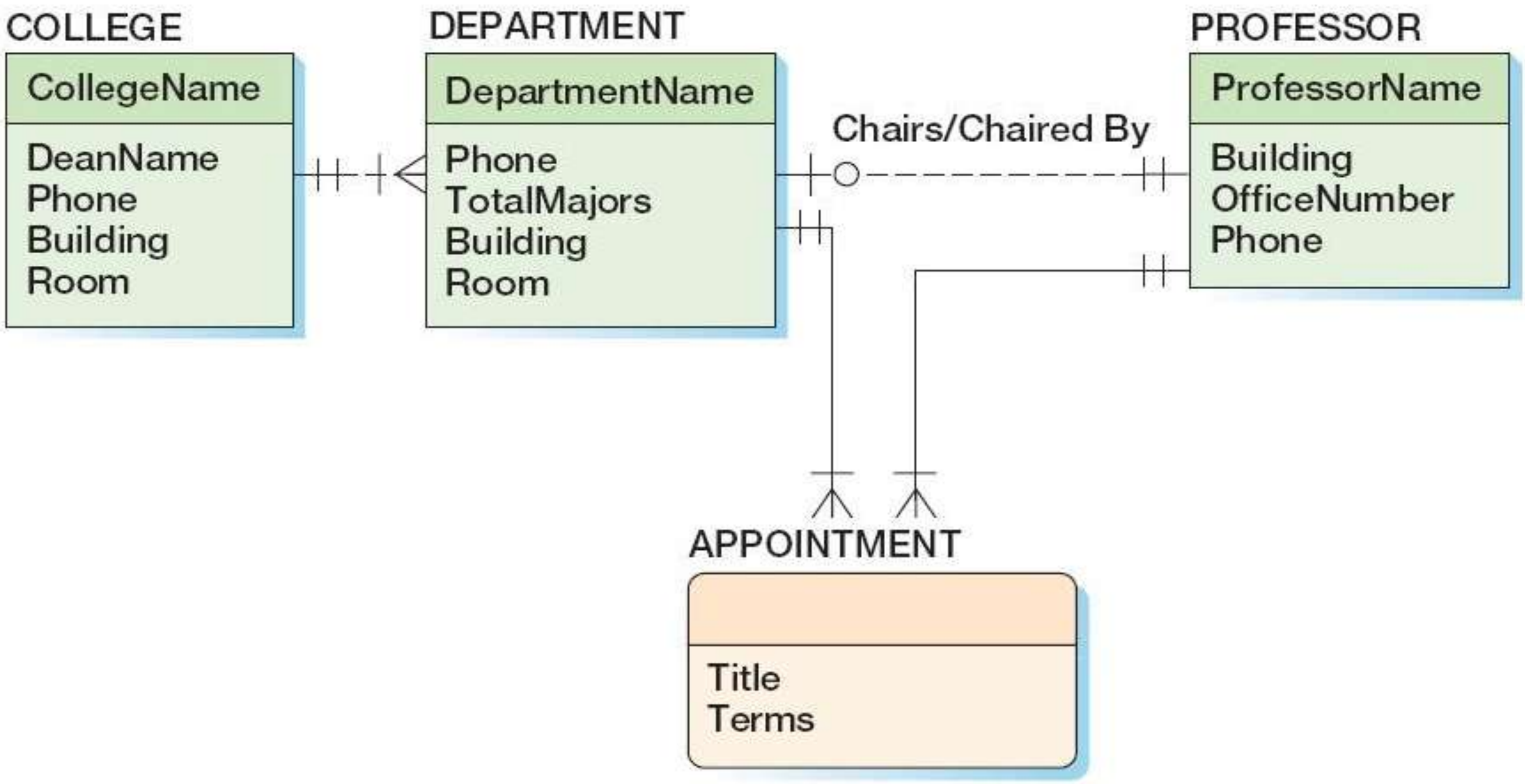
- Each appointment letter associates a professor and department, and has details regarding such appointment.

- An "association pattern" is designed to model these appointments, and is added to the E-R model.



# Making the Chairperson a Relationship

- Since a chairperson is also a professor in a department, we can improve the E-R model by removing the chairperson data from the DEPARTMENT entity set, and replace it with a chairperson relationship.

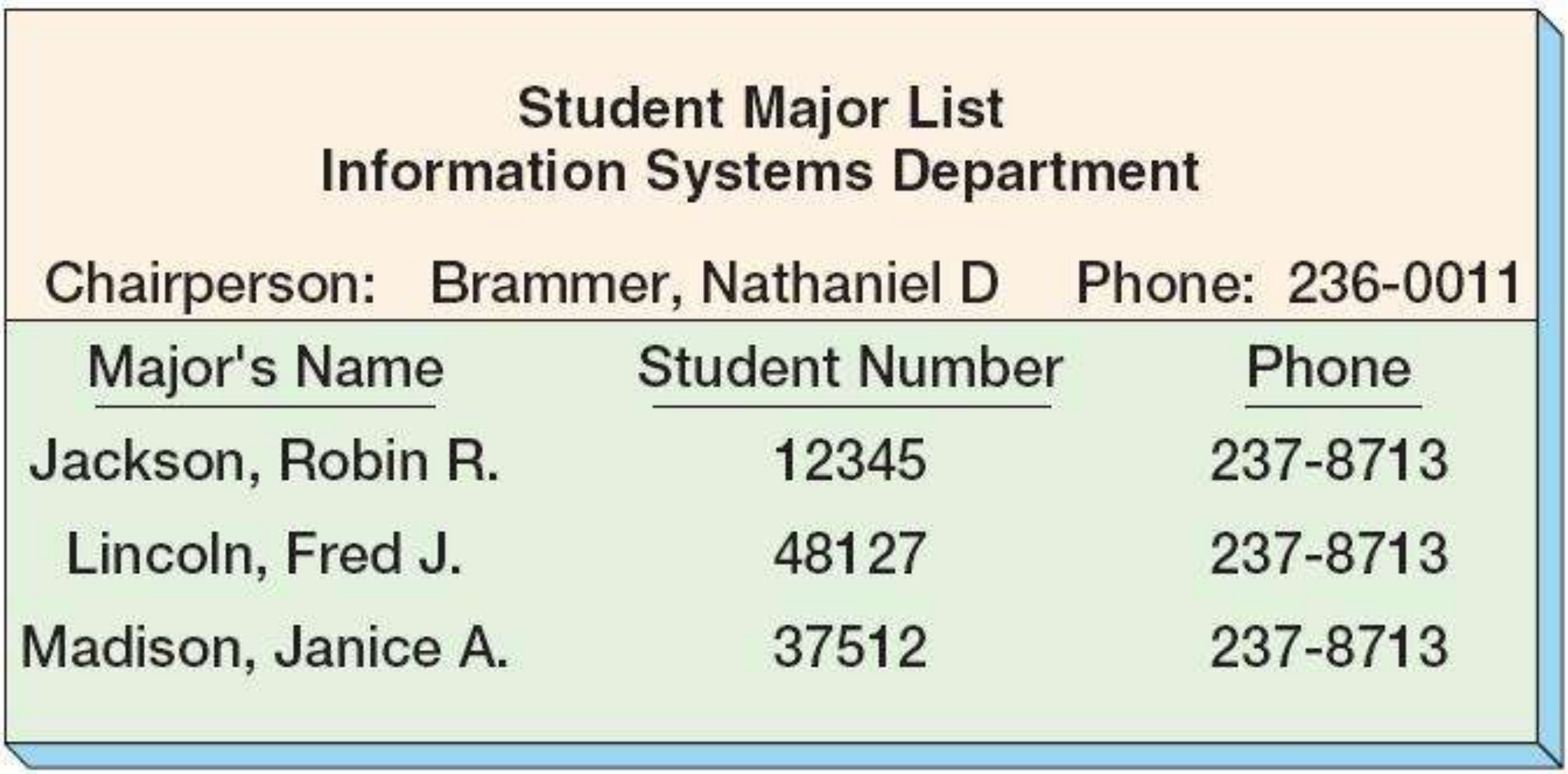


# The Chairperson Relationship

- Notice that a given entity set can be involved in several relationships.
- By modeling chairperson as a relationship, and removing the chairperson attribute from the DEPARTMENT entity set, we provide some additional flexibility for our E-R model.
  - For instance, we can now model situations at Highline where the chairperson has an office location and phone different from the Department office and phone.
- Notice also that the min and max cardinality state that a professor can chair 0 or 1 departments.
  - So, the chairperson chairs 1 department, and the other professors chair 0 departments.
- Can you think of any other way to model which professor serves as chairperson?

# The Department/Major Report

- The following is an existing report from the Department of Information Systems, and we are told that every academic department at Highline University produces a similar report enumerating its student majors.



**Student Major List**
**Information Systems Department**

Chairperson: Brammer, Nathaniel D Phone: 236-0011

| Major's Name | Student Number | Phone |
|---|---|---|
| Jackson, Robin R. | 12345 | 237-8713 |
| Lincoln, Fred J. | 48127 | 237-8713 |
| Madison, Janice A. | 37512 | 237-8713 |

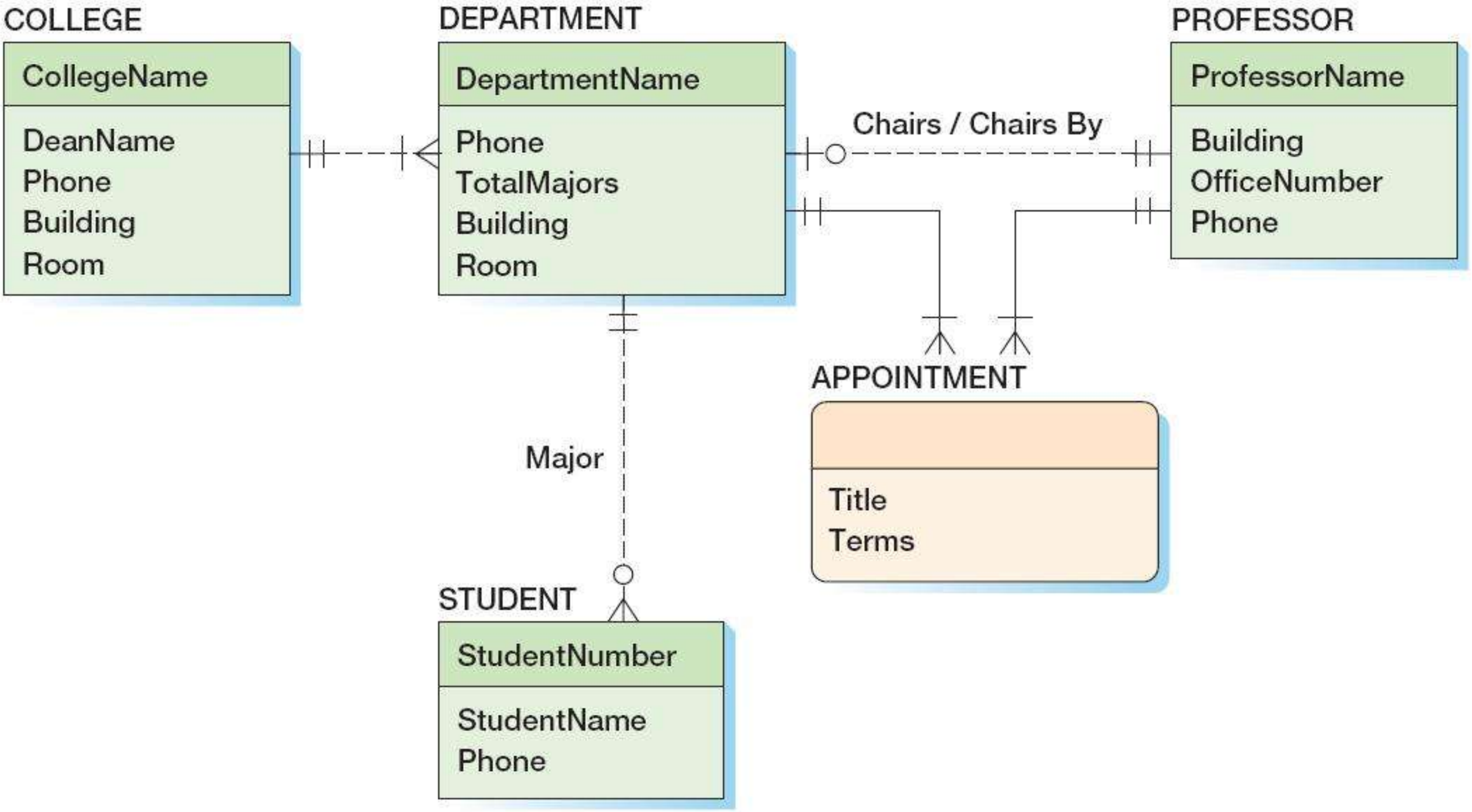
# Extracting From The Department Report

- We determine that there should be an entity set STUDENT in our E-R model, having attributes like StudentNumber, StudentName and Phone.

- Students are not ID-dependent on Departments, so we have a non-identifying relationship between STUDENT and DEPARTMENT (and denoted by a dashed line).

- Interviews with the college administration reveal that every student must have one (and only one) declared major, and that there could be a major having no students, though most majors have “many” students.

# E-R Model With The STUDENT Entity Set



COLLEGE

CollegeName

DeanName
Phone
Building
Room

DEPARTMENT

DepartmentName

Phone
TotalMajors
Building
Room

Chairs / Chairs By

PROFESSOR

ProfessorName

Building
OfficeNumber
Phone

APPOINTMENT

Title
Terms

Major

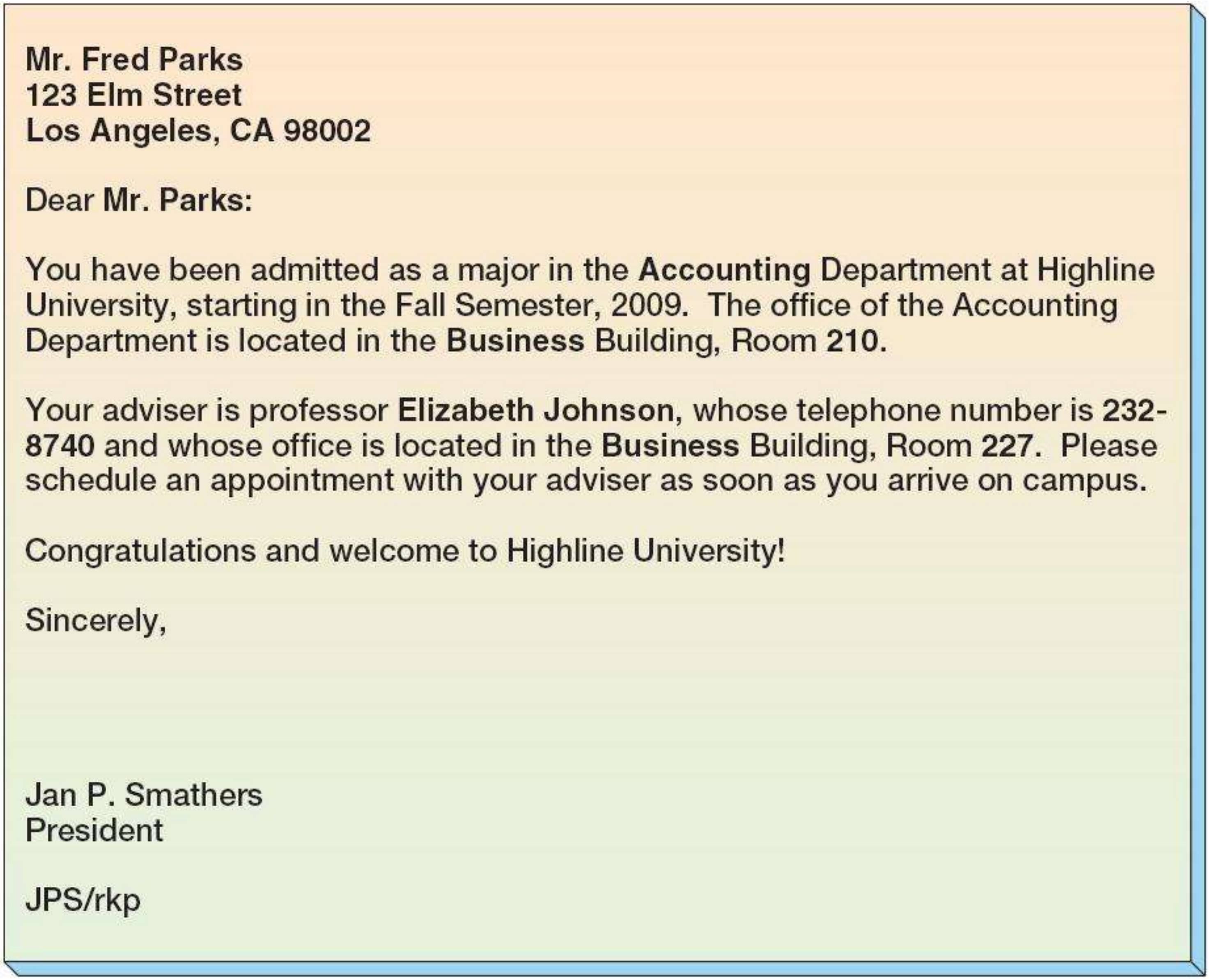
STUDENT

StudentNumber

StudentName
Phone

# The Student Admittance Letter

- Highline University sends to every admitted student a letter, such as this letter to Mr. Fred Parks:



**Mr. Fred Parks**
**123 Elm Street**
**Los Angeles, CA 98002**

Dear **Mr. Parks:**

You have been admitted as a major in the **Accounting** Department at Highline University, starting in the Fall Semester, 2009. The office of the Accounting Department is located in the **Business** Building, Room **210**.

Your adviser is professor **Elizabeth Johnson**, whose telephone number is **232-8740** and whose office is located in the **Business** Building, Room **227**. Please schedule an appointment with your adviser as soon as you arrive on campus.

Congratulations and welcome to Highline University!

Sincerely,

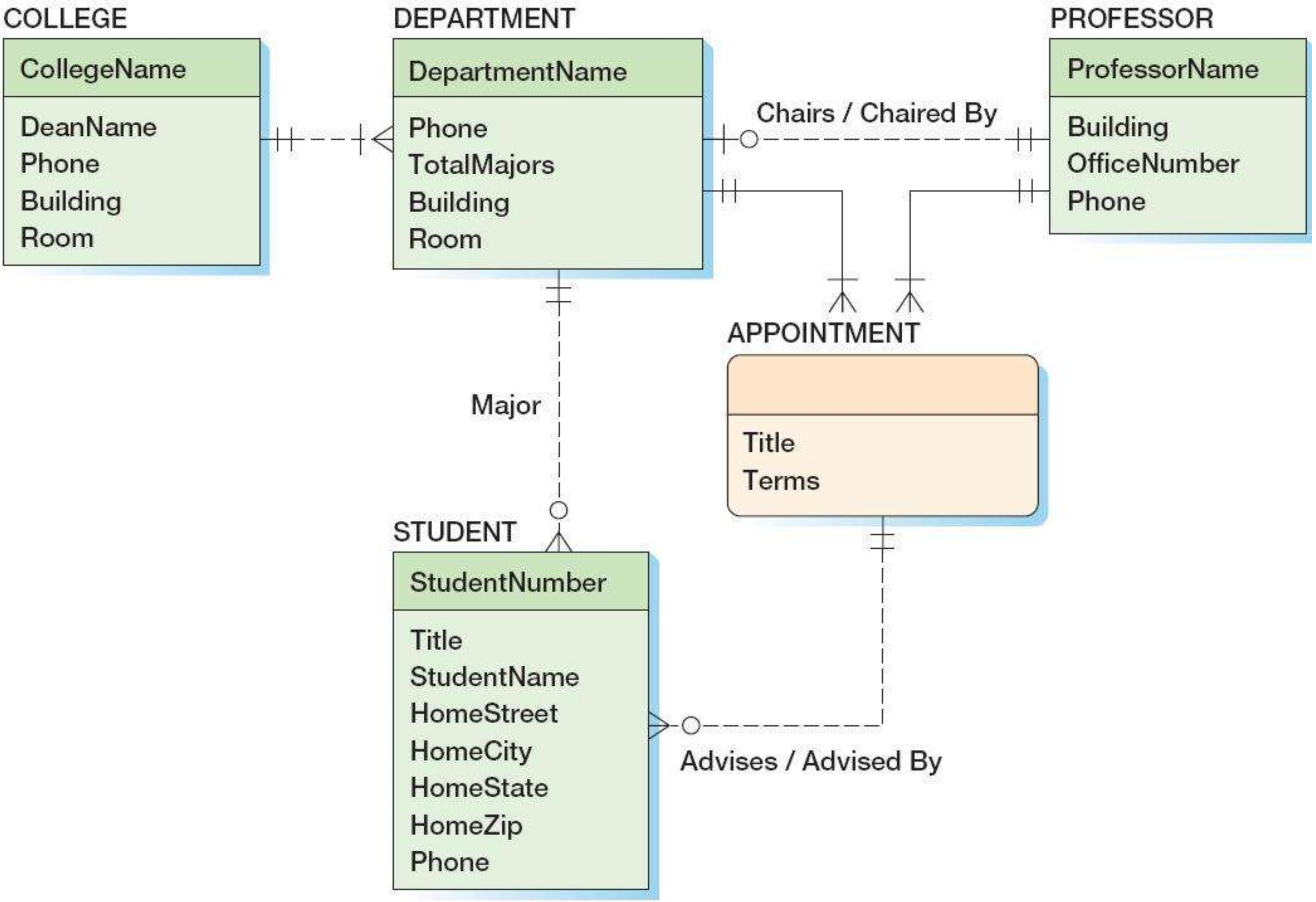
Jan P. Smathers
President

JPS/rkp

# Extracting from the Student Admittance Letter

- From the Admittance letter, we see that there is additional information relating to the student (such as the student's home address) and his/her Department Advisor.
- Interviews with the College Administration reveal that
  - the Department Advisor is always some professor in the student's major department
  - a student is **required** to have an Advisor
  - some professors will serve as student Advisors, while other professors will not serve as student Advisors
- The Admittance Letter uses "Mr" so we have added an attribute to the STUDENT entity set called "title" (which will either be "Mr" or "Ms"), though we also could have added an attribute "gender"
- We can establish a relationship between STUDENT and APPOINTMENT (since appointment will give us the requisite data about the Professor who is serving as the Advisor to the Student)
  - Alternatively, we could have established a relationship between STUDENT and PROFESSOR.

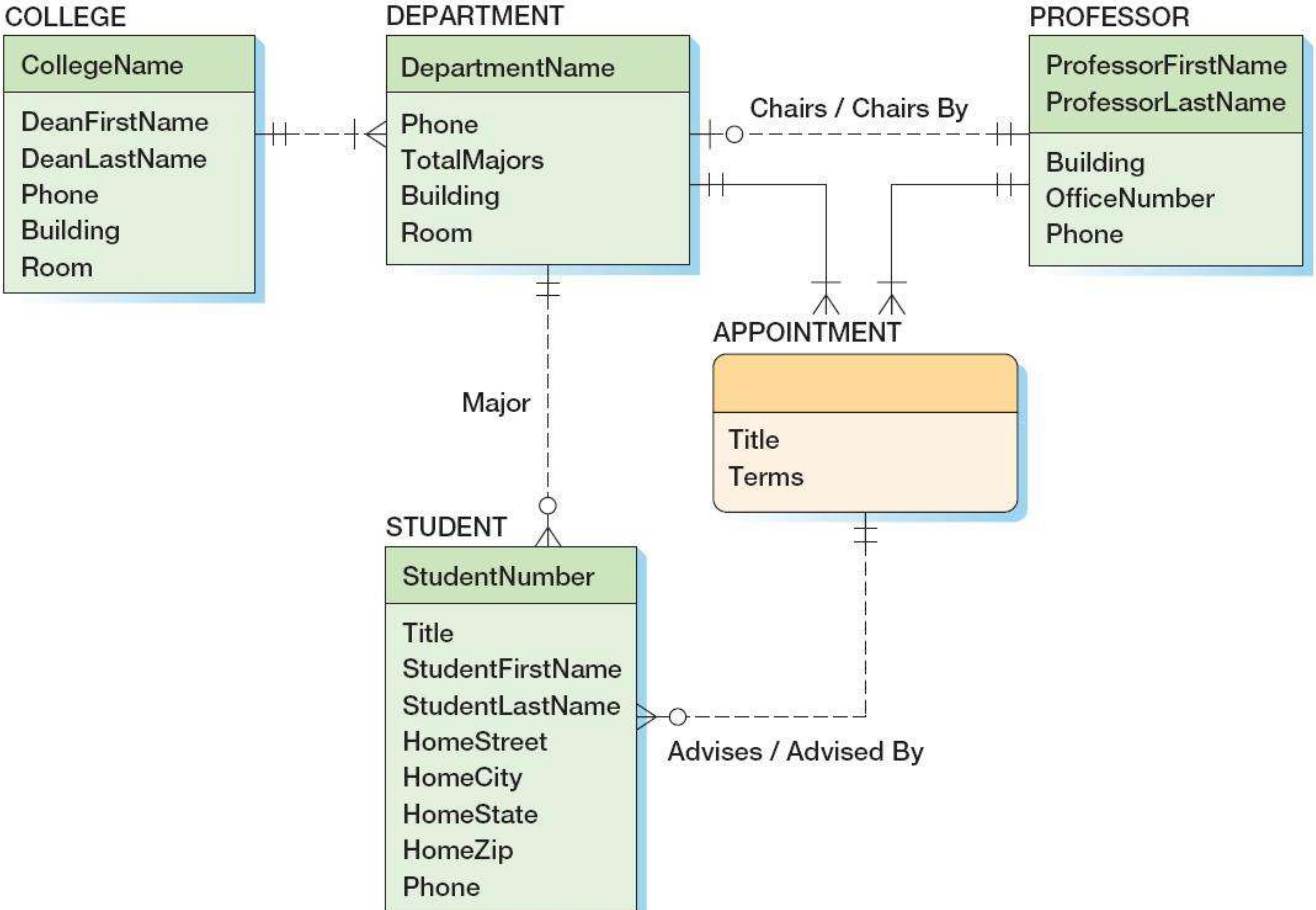
# The E-R Model With The Advises Relationship



# Slight Refinement to the E-R Model

- From the Admittance Letter, we see that we need to not have just the fullname of each student, but we should also store both their first name and last name.
  - We decide that all names, including the names of professors and Deans, should be stored as two attributes: a firstname, and a lastname
- This leads us to the next (and (sic) “final”) iteration of the E-R data model for Highline University
- The iterative process we have gone through, of extracting needed entity sets, attributes and relationship from existing reports, forms, letters and interviews, is typical of the process by which we build a data model.

# The Resulting E-R Model for Highline University



COLLEGE

CollegeName

DeanFirstName
DeanLastName
Phone
Building
Room

DEPARTMENT

DepartmentName

Phone
TotalMajors
Building
Room

PROFESSOR

ProfessorFirstName
ProfessorLastName

Building
OfficeNumber
Phone

Chairs / Chairs By

APPOINTMENT

Title
Terms

Major

STUDENT

StudentNumber

Title
StudentFirstName
StudentLastName
HomeStreet
HomeCity
HomeState
HomeZip
Phone

Advises / Advised By