Conceptual Modeling

(Crow's Foot Notation)

TEXT: CHAPTER 5

Agenda

Entities
- Know: what they are, how do we represent them

Relationships
- Types of relationships (binary, ternary, etc.)

Conceptual Modeling
- Chen's Entity-Relationship (E-R) Model
- Crows Foot
  - Example: Highline University (from text)

The Entity-Relationship (E-R) Model

- **Entity-Relationship (E-R) model** is a set of concepts and graphical symbols that can be used to create conceptual models (sometime called schemas).
- Versions:
  - **Original E-R model**— Peter Chen (1976) MIT
  - **Extended E-R model**—later extensions to Chen model included is-a relationships, etc.
    - Referred to as Extended E-R model.

For this class: We will call this *Chen's Notation (Original or extended with subclasses)*

Important: Why study conceptual models?

- Means of communication between database designer and users.
- Represents real world application for which a database is needed before implementation.

Entity-Relationship Model: Versions

- **Original E-R model**—by Peter Chen (1976)
- **Extended E-R model**—extensions to the Chen model adding sub/super classes
- **Information Engineering (IE)**—by James Martin (1990); uses "crow's foot" notation, in text
- **IDEF1X**—a national standard developed by the National Institute of Standards and Technology [see Appendix C] [*Not for this class.]*
- **Unified Modeling Language (UML)**—by the Object Management Group; supports object-oriented methodology [see Appendix D] [*Not for this class.*]

For this class (2 versions):

Chen's notation

Crow's foot / Crow's feet notation

Note: Either acceptable for assignments/exams.

Avoid mixing notations

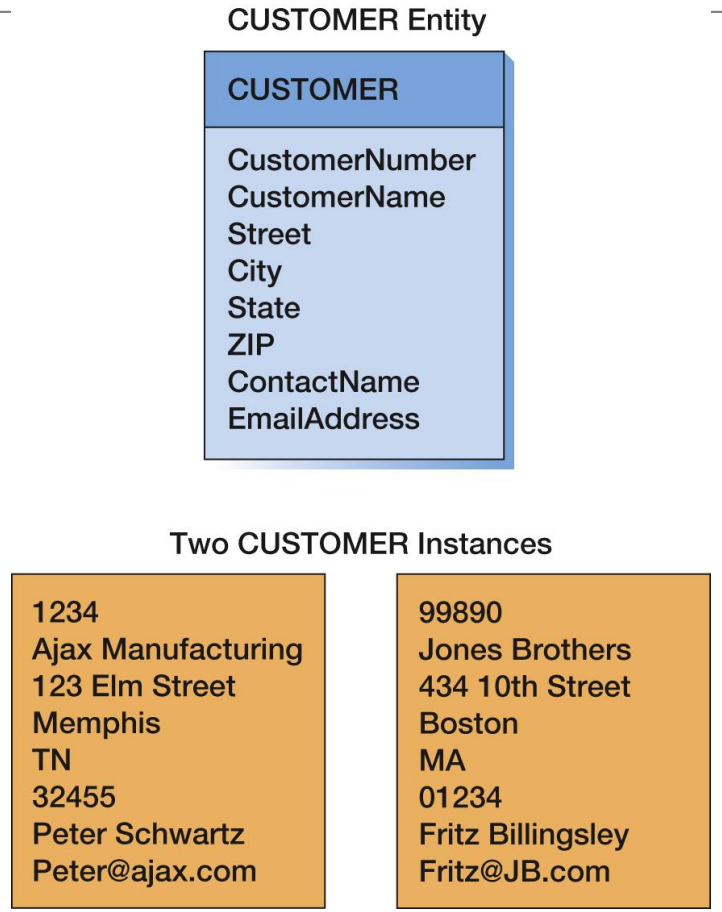
Entities

- Something that can be readily identified and that users want to track:
  - **Entity class**—a collection of entities of a given type
  - **Entity instance**—the occurrence of a particular entity

    [Terminology: instance / occurrence / instantiation]
- There are usually many instances of an entity in an entity class.
  - How many? Depends on application.

Note: entity class, entity type, or just entity. For class, use “entity.”

Figure 5-1: Crow's Foot Notation
CUSTOMER Entity and Two Entity Instances



Attributes

Attributes describe an entity's characteristics.

All entity instances of a given entity class have the same attributes, but vary in values.

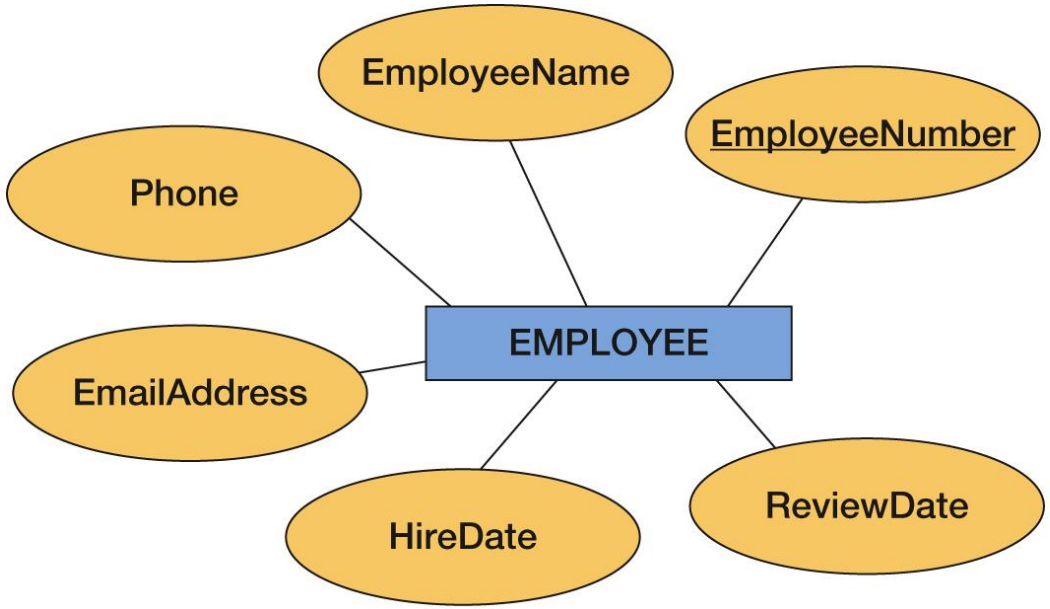
- E.g., all instances of the class student have an attribute, student-name, but the values will vary.

Shown in data models as **ellipses**.

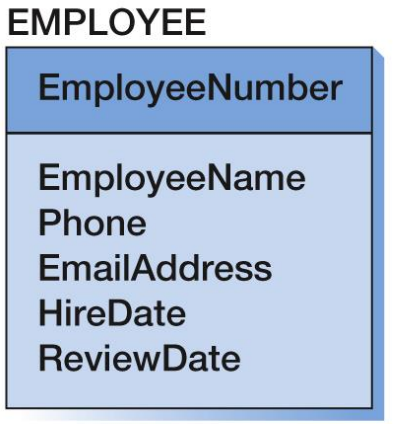
Data modeling products today commonly show attributes in **rectangular form**.

- (E.g., Crow's Feet notation)
- Note: the two different representations

Figure 5-2 Variations of Attributes with E-R Models



(a) Attributes in Ellipses



(b) Attributes in Rectangle

Attributes in ellipses: Chen Model

Attributes in rectangle: Crow's feet

Identifiers

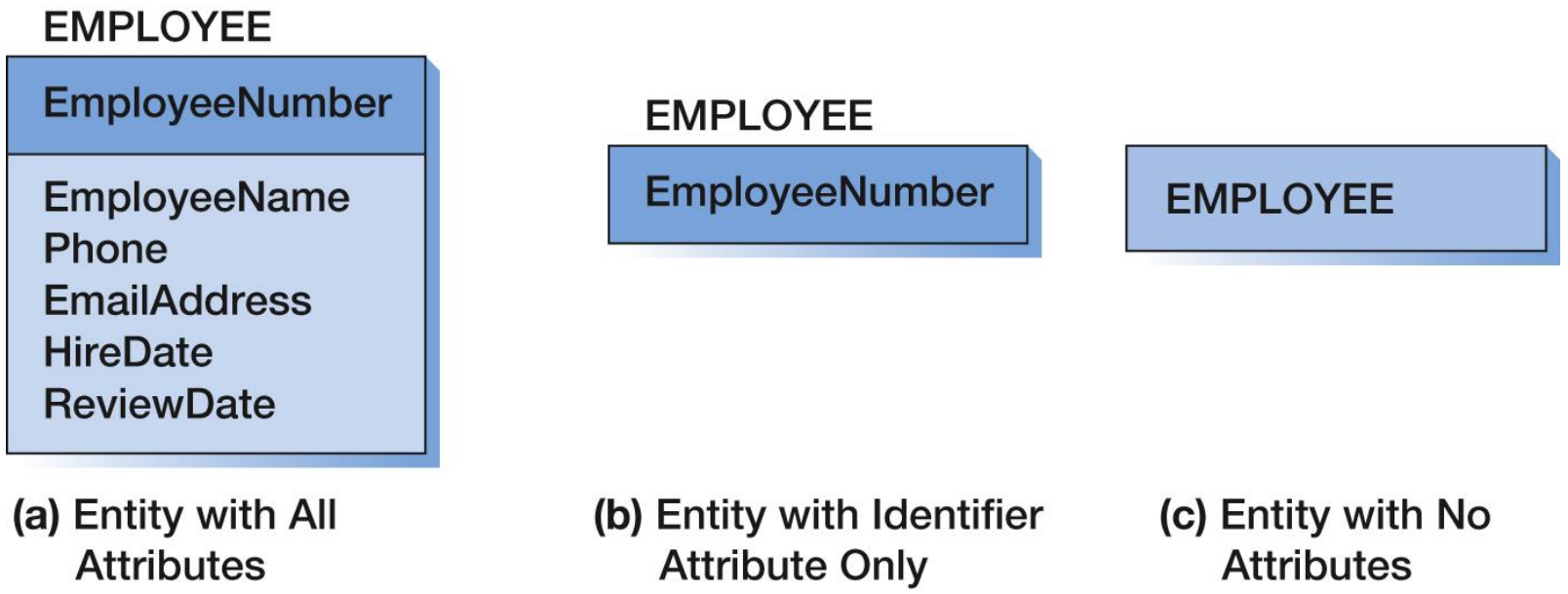
Identifiers are attributes that name, or identify, entity instances.

Composite identifiers are identifiers that consist of two or more attributes.

Identifiers become keys.

- Entities have identifiers/keys.
- Relationships do not have keys.
- Tables (or relations) have keys.
- [*Note*: Do not get these confused. More explanation later.]

Figure 5-3 Entity and Attributes



(a) Entity with All Attributes

(b) Entity with Identifier Attribute Only

(c) Entity with No Attributes

Starting to identify entities. Conceptual models must have minimally a key attribute

Relationships

Entities associated with one another in **relationships**:

- **Relationship classes**: associations among entity classes
- **Relationship instances**: associations among entity instances

A relationship class can involve two or more entity classes.

Note from the authors: In the original E-R model, relationships could have attributes, but today this is no longer done.

Note from instructor: Depends on modeling choice. Examples coming.

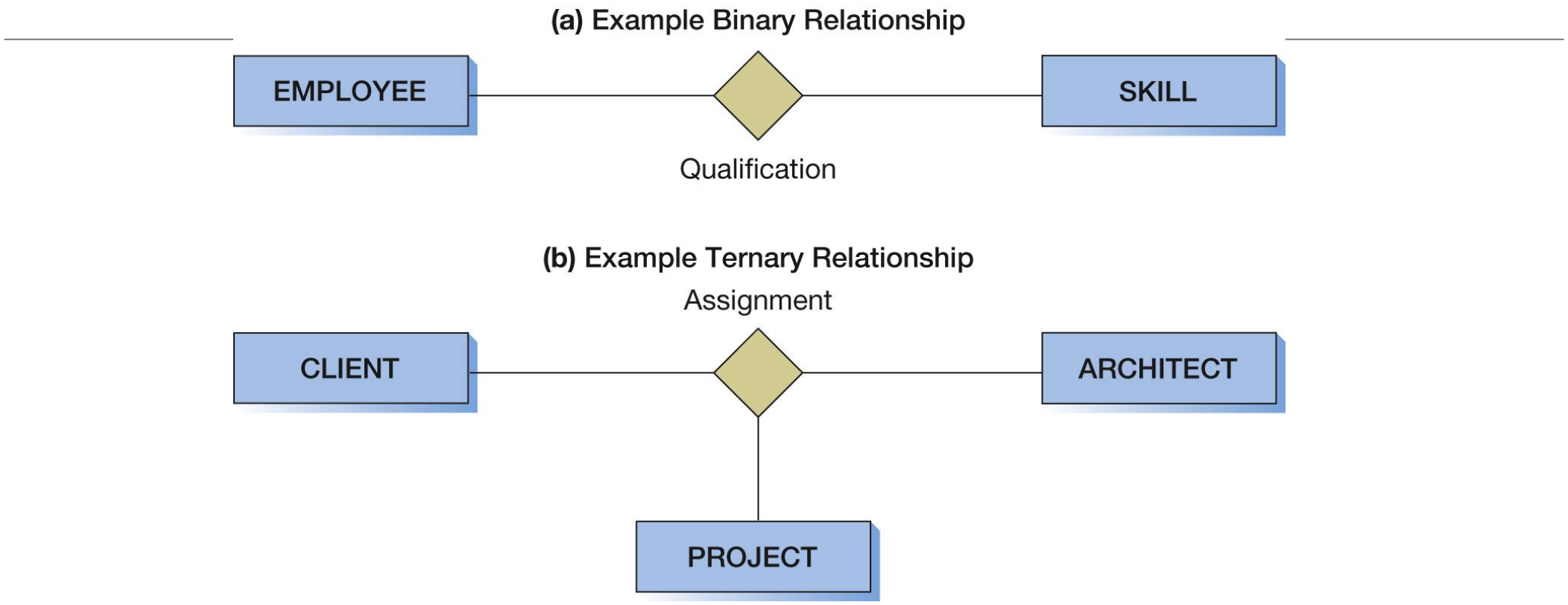
Important concept: Student takes Course and receives grade

Degree of the Relationship

The **degree** of the relationship is the number of entity classes in the relationship:

- Two entities have a **binary relationship** of degree two.
- Three entities have a **ternary relationship** of degree three.

Figure 5-4 Binary Versus Ternary Relationships



Interpretation?

Entities and Tables

- From text: The principle difference between an **entity** and a **table (relation)** is that you can express a relationship between entities *without using foreign keys*.
  - Not exactly. An entity is a conceptual modeling construct.
  - A table (relation) occurs at the logical phase of database design. A table will represent an entity. More on this during logical design.
- Emphasis on the conceptual modeling phase.
  - This mixes the phrases, so do not be confused.
  - More on this later.

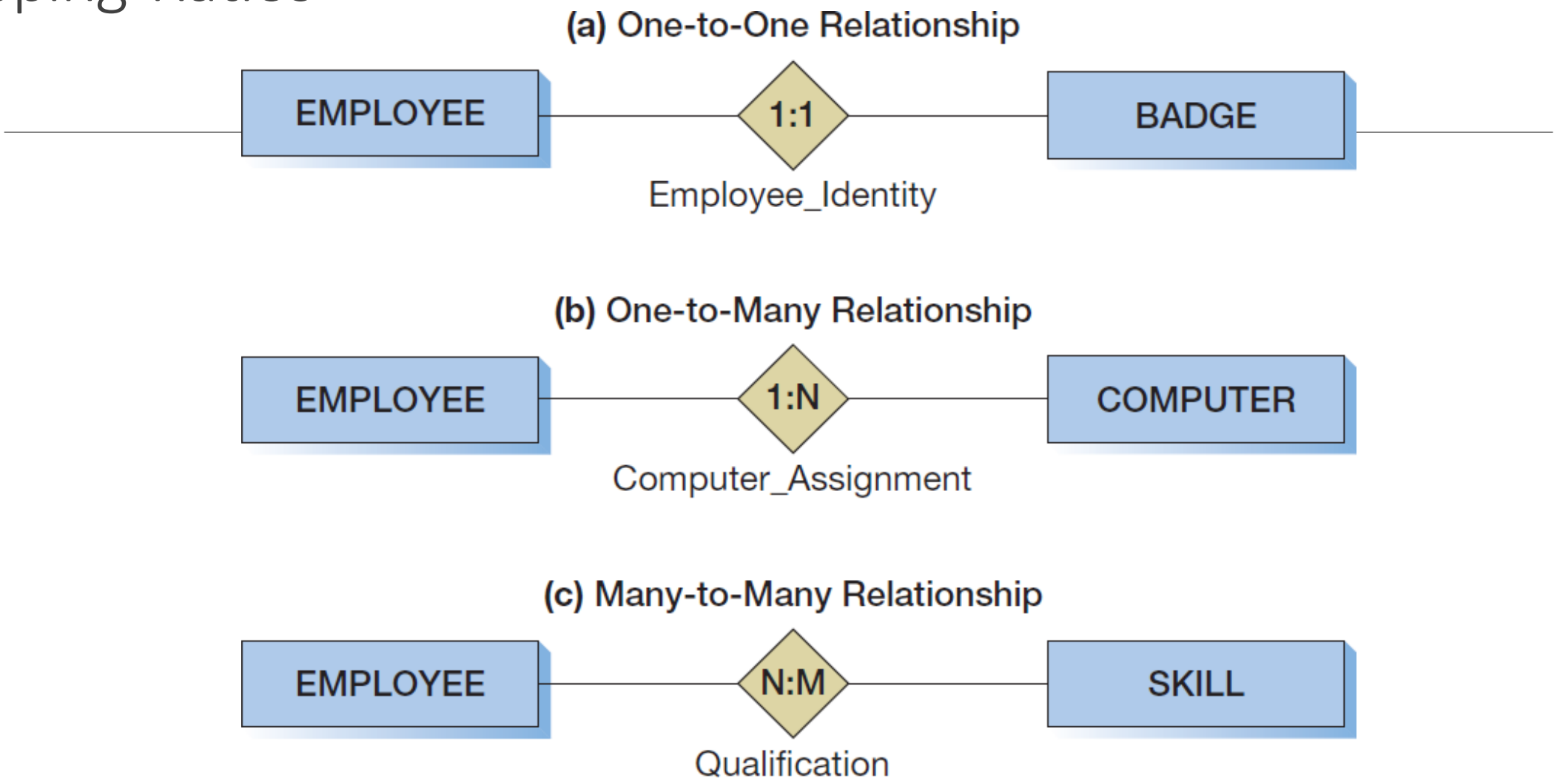
Cardinality

Cardinality means “count,” and is expressed as a number.

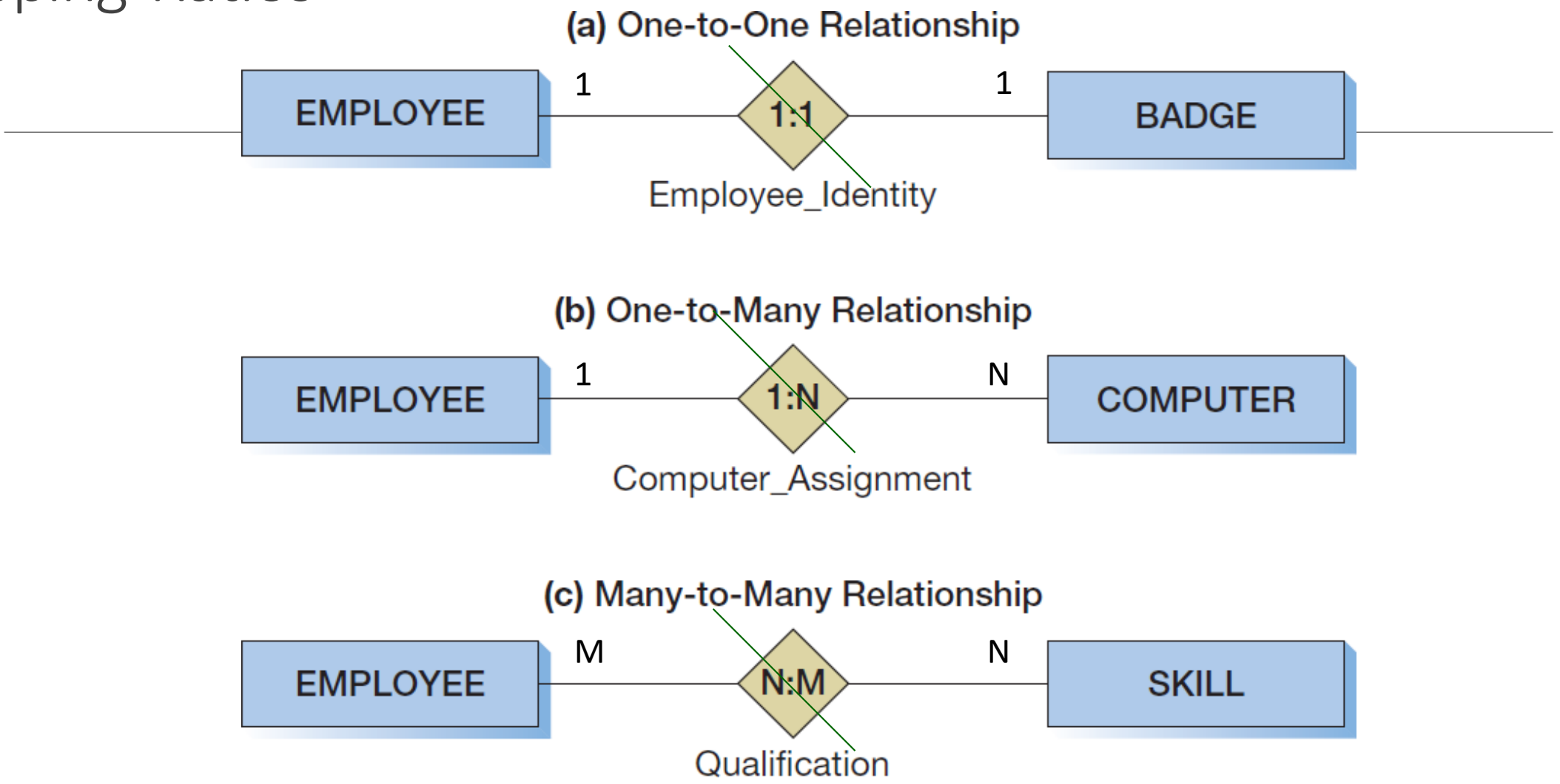
Maximum cardinality is the maximum number of entity instances that ***can (allowed to)*** participate in a relationship.

Minimum cardinality is the minimum number of entity instances that ***must (required)*** participate in a relationship.

Mapping Ratios



Mapping Ratios



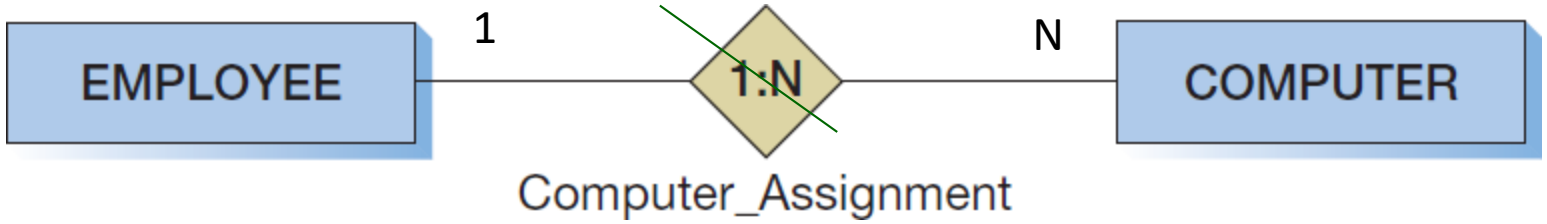
Parent and Child Entities

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**.

Note: ok to think of it this way, but provided mapping ratios are correct, that is enough.

EMPLOYEE is the parent and COMPUTER is the child:



Min/Max Cardinalities

Minimum Cardinality

Minimum cardinality is the minimum number of entity instances that ***must*** participate in a relationship.

Minimums are generally stated as either **zero** or **one**:

- IF **zero [0]** THEN participation in the relationship by the entity is **optional**, and ***no*** entity instance must participate in the relationship.
- IF **one [1]** THEN participation in the relationship by the entity is **mandatory**, and ***at least one*** entity instance must participate in the relationship.

[Notes: Finer level of detail coming.

Optionality is an important concept.]

Indicating Minimum Cardinality

Crow's Feet Notation:

- **Minimum cardinality of zero [0]** indicating ***optional*** participation is indicated by placing an **oval** next to the optional entity.
- **Minimum cardinality of one [1]** indicating ***mandatory (required)*** participation is indicated by placing a **vertical hash mark** next to the required entity.

** There are multiple notations. Need to understand this for Crow's Feet.

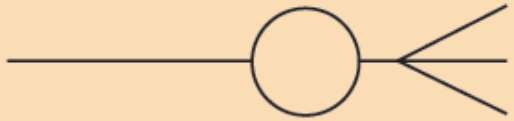
Reading Minimum Cardinality

Look toward the entity in question [note direction]:

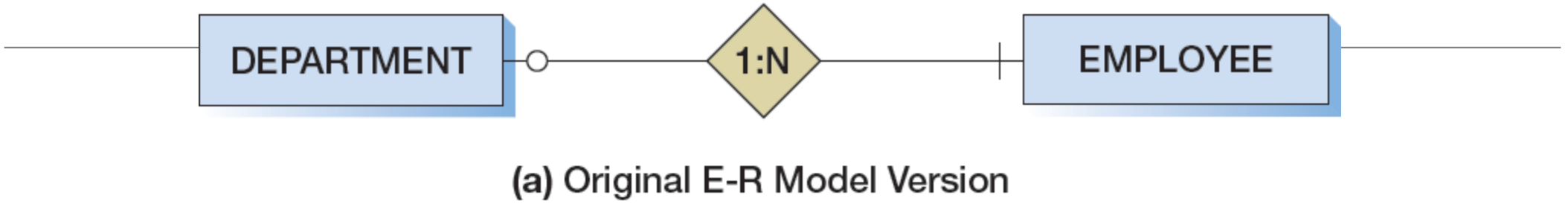
- IF you see an **oval** THEN that entity is **optional** (minimum cardinality of zero [0]).
- IF you see a **vertical hash mark** THEN that entity is **mandatory** (required) (minimum cardinality of one [1]).

Note: Optional versus mandatory is important and has implications on the final design of a database and the enforcement of integrity.

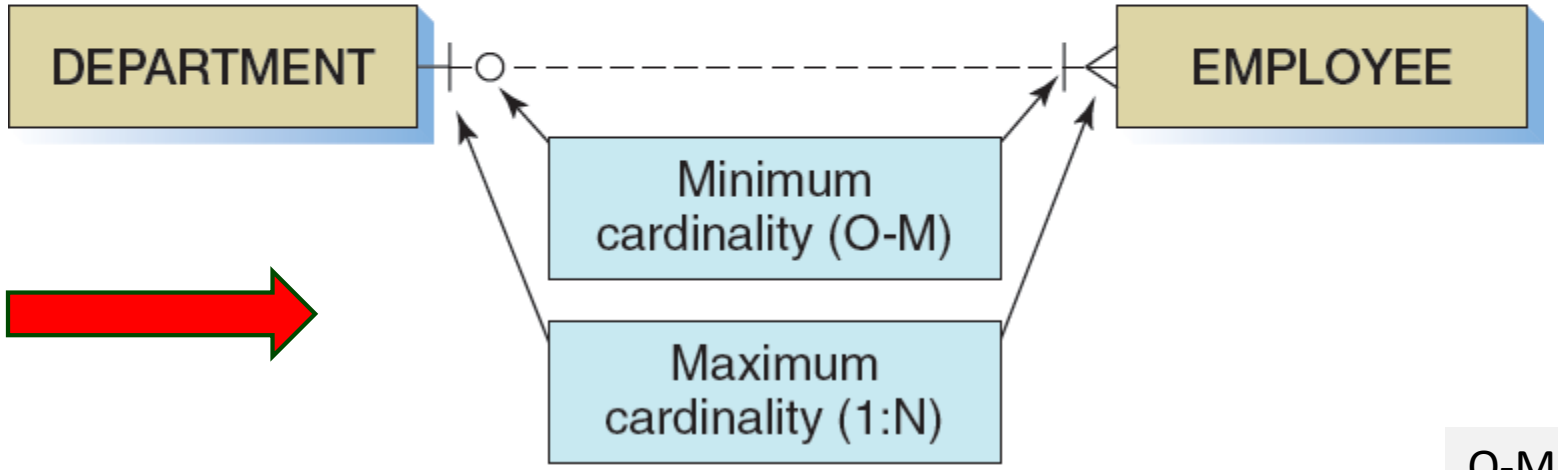
Data Modeling Notation: IE Crow's Foot

Symbol	Meaning	Numeric Meaning
	Mandatory—One	Exactly one
	Mandatory—Many	One or more
	Optional—One	Zero or one
	Optional—Many	Zero or more

Data Modeling Notation: IE Crow's Foot 1:N



(a) Original E-R Model Version

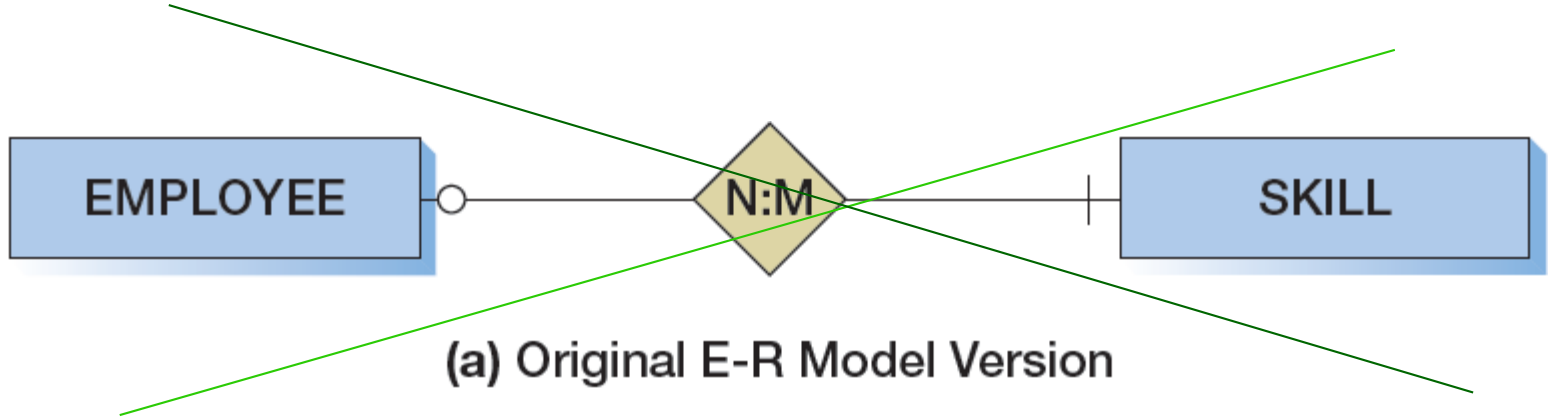


(b) Crow's Foot Version

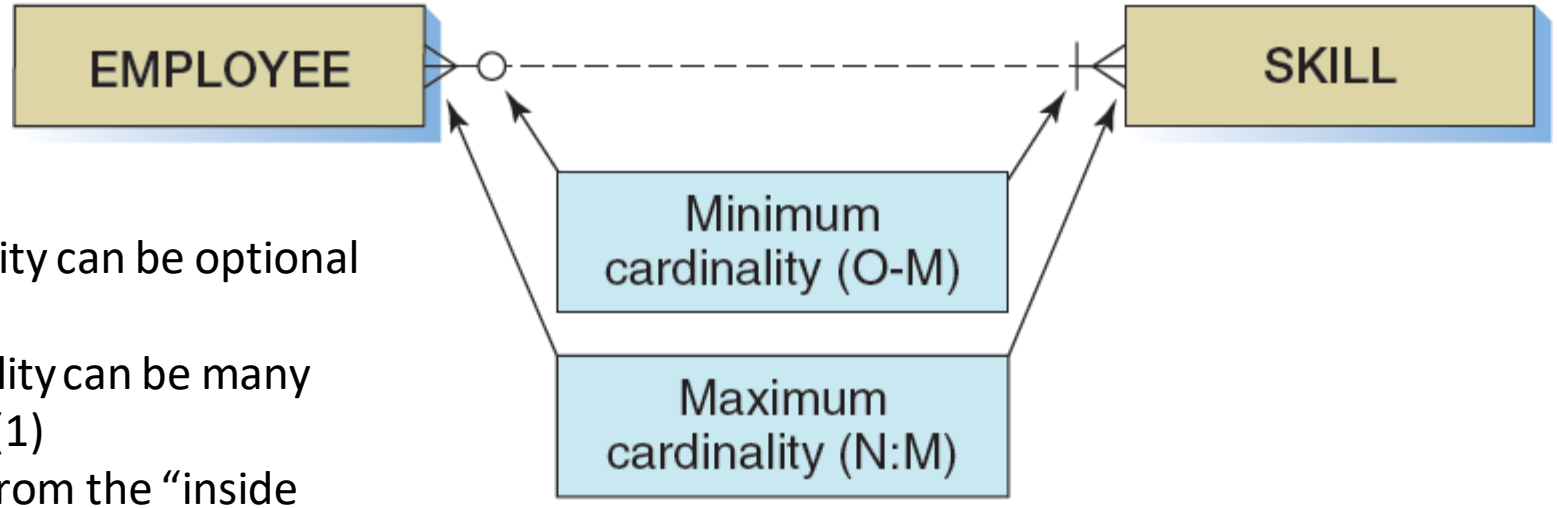
O-M stands for Optional-Mandatory

Data Modeling Notation: IE Crow's Foot N:M

Ignore



(a) Original E-R Model Version



(b) Crow's Foot Version

Note:

- Minimum cardinality can be optional or mandatory
- Maximum cardinality can be many (N) or mandatory (1)
- These are "read" from the "inside out."

ID-Dependent Entities

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

The ID-dependent entity is a logical extension or subunit of the parent:

- **BUILDING : APARTMENT**
- **PAINTING : PRINT**

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

Strong and Weak Entities

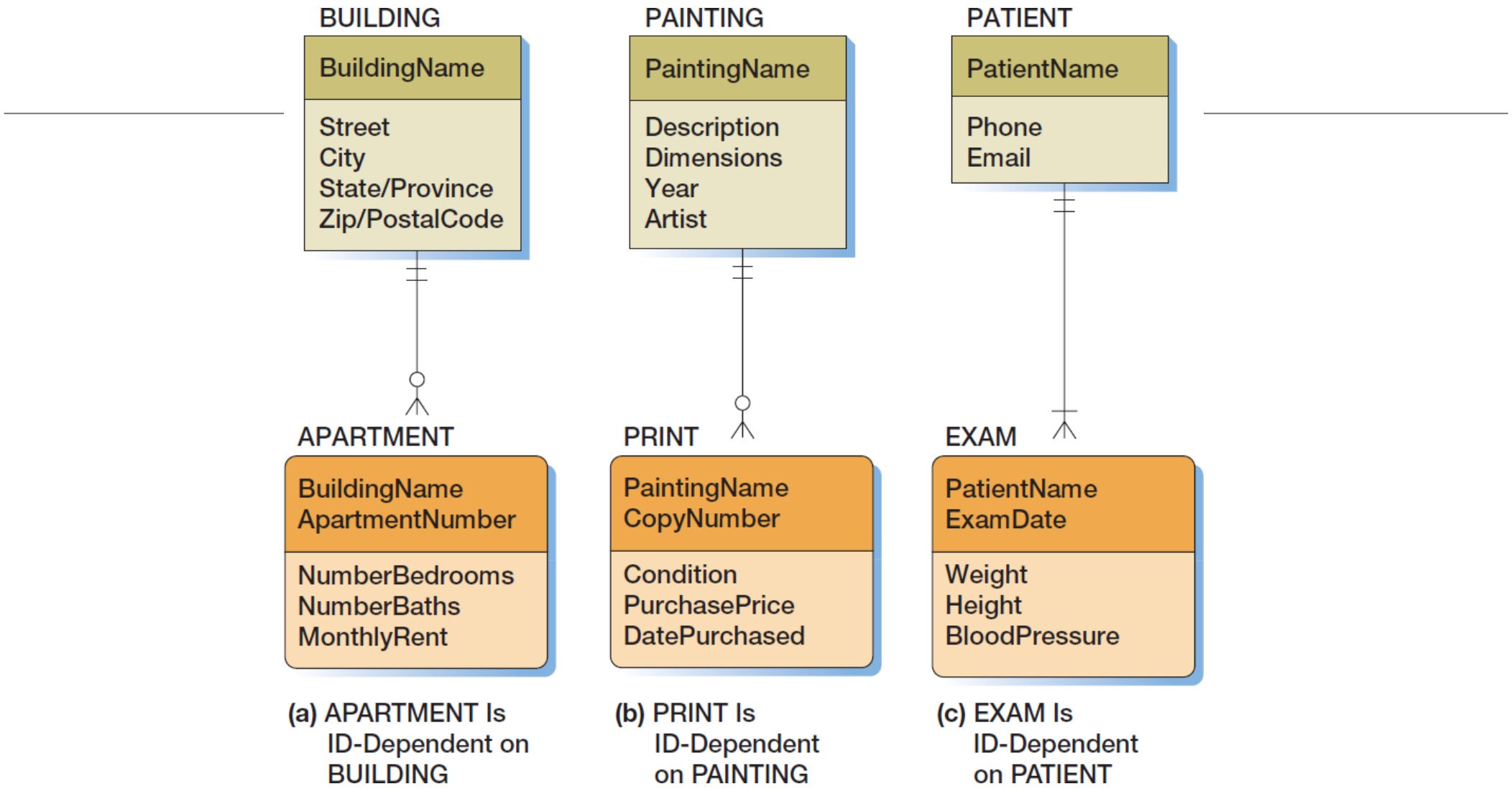
A **Strong Entity** is an entity that represents something that can exist on its own.

- Examples (PERSON, AUTOMOBILE, BUILDING)

A **Weak Entity** is an entity whose existence depends on the presence of another entity.

- Example (APARTMENT – depends on BUILDING)

ID-Dependent Entities



(a) APARTMENT Is ID-Dependent on BUILDING

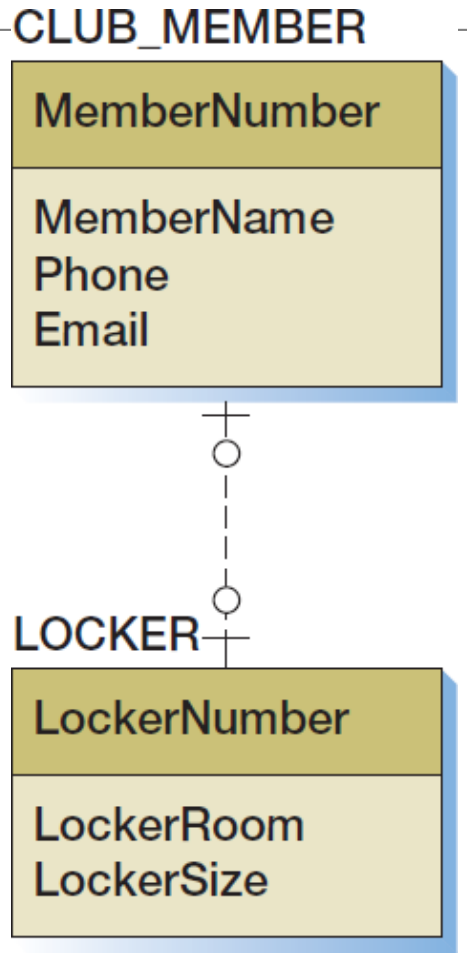
(b) PRINT Is ID-Dependent on PAINTING

(c) EXAM Is ID-Dependent on PATIENT

Strong Entity Patterns:

1:1 Strong Entity Relationships

[Most important to understand the min/max cardinalities. Not required to refer to these as "strong" for the purposes of this course]



Interpretation?

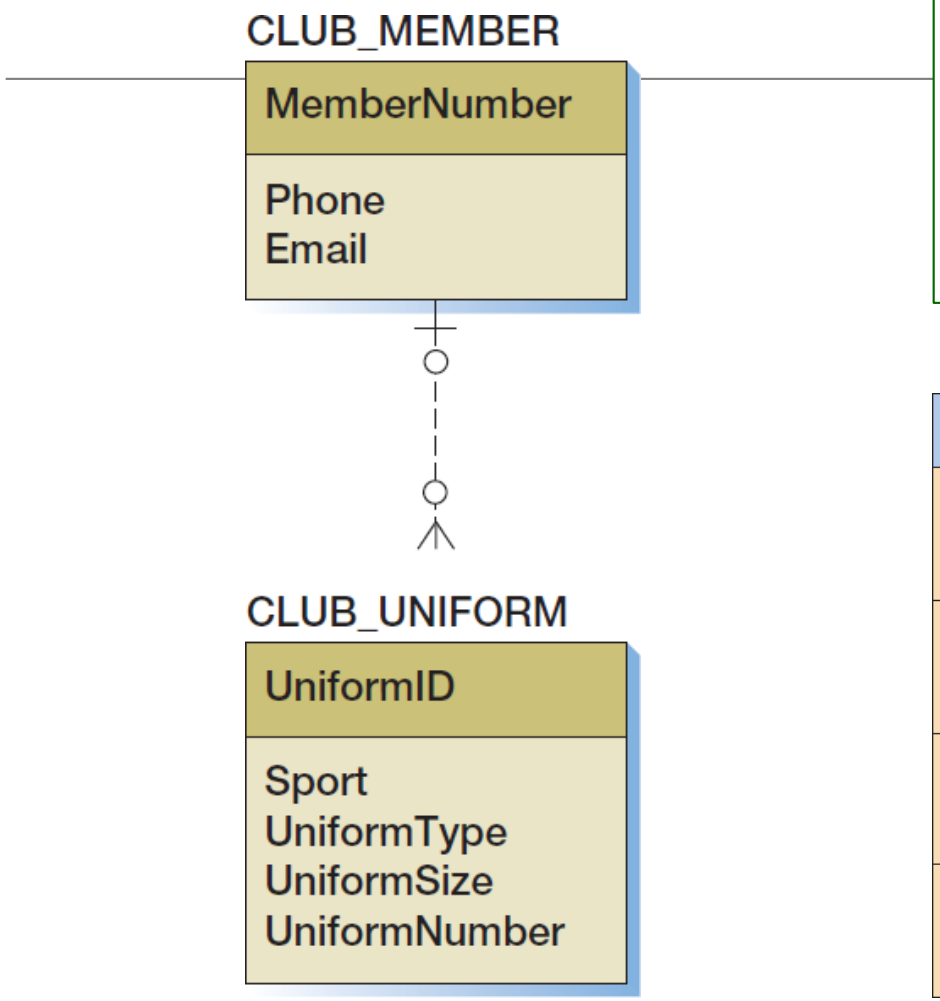
Corresponding business rules?

Recall:

Symbol	Meaning	Numeric Meaning
	Mandatory—One	Exactly one
	Mandatory—Many	One or more
	Optional—One	Zero or one
	Optional—Many	Zero or more

Note: Optional

Strong Entity Patterns:
1:N Strong Entity Relationships



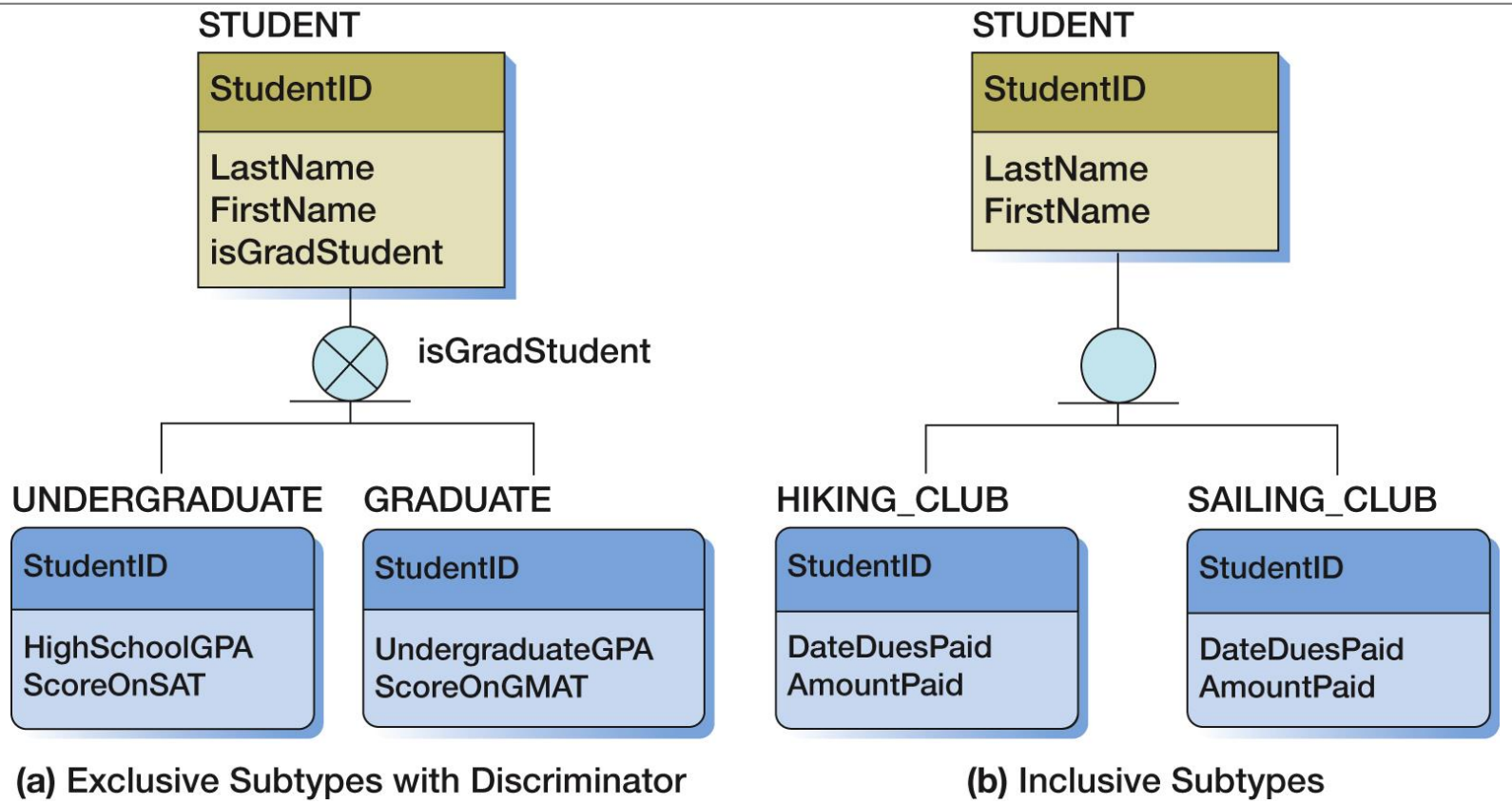
Real-world interpretation: A club member can have more than one club uniform, but a uniform can only belong to one club member.

Corresponding business rules?

Recall:

Symbol	Meaning	Numeric Meaning
	Mandatory—One	Exactly one
	Mandatory—Many	One or more
	Optional—One	Zero or one
	Optional—Many	Zero or more

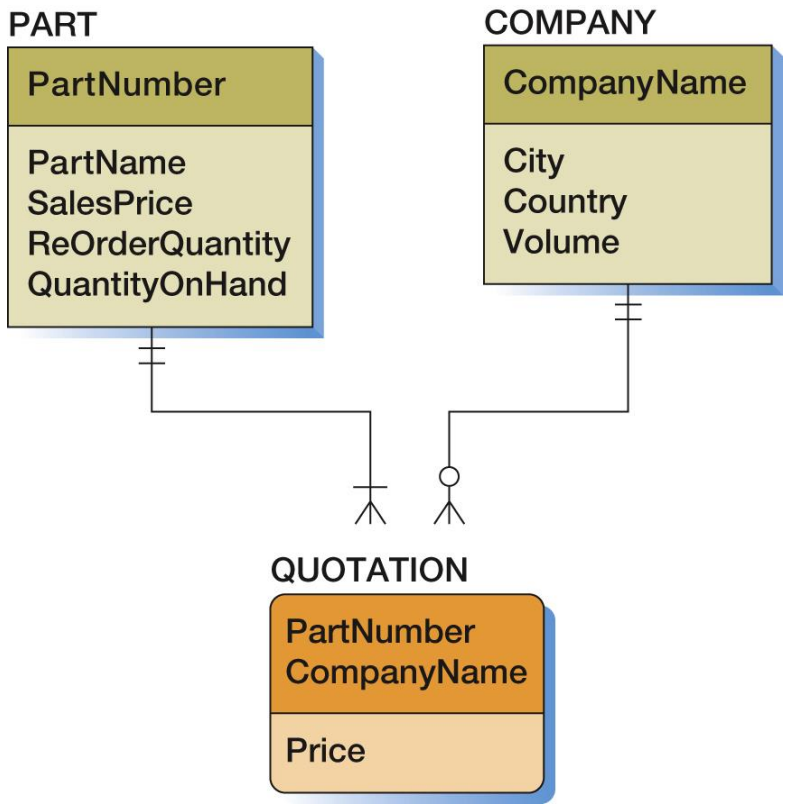
Figure 5-13 Examples of Subtype Entities



(a) Exclusive Subtypes with Discriminator

(b) Inclusive Subtypes

Figure 5-22 Association Pattern for Report in Figure 5-21 [Many to many]



Draw the corresponding entity-relationship model with Chen's notation.

Note: relationship attribute

Figure 5-23 Association Pattern for Tenary Relationship in Fig. 5-4

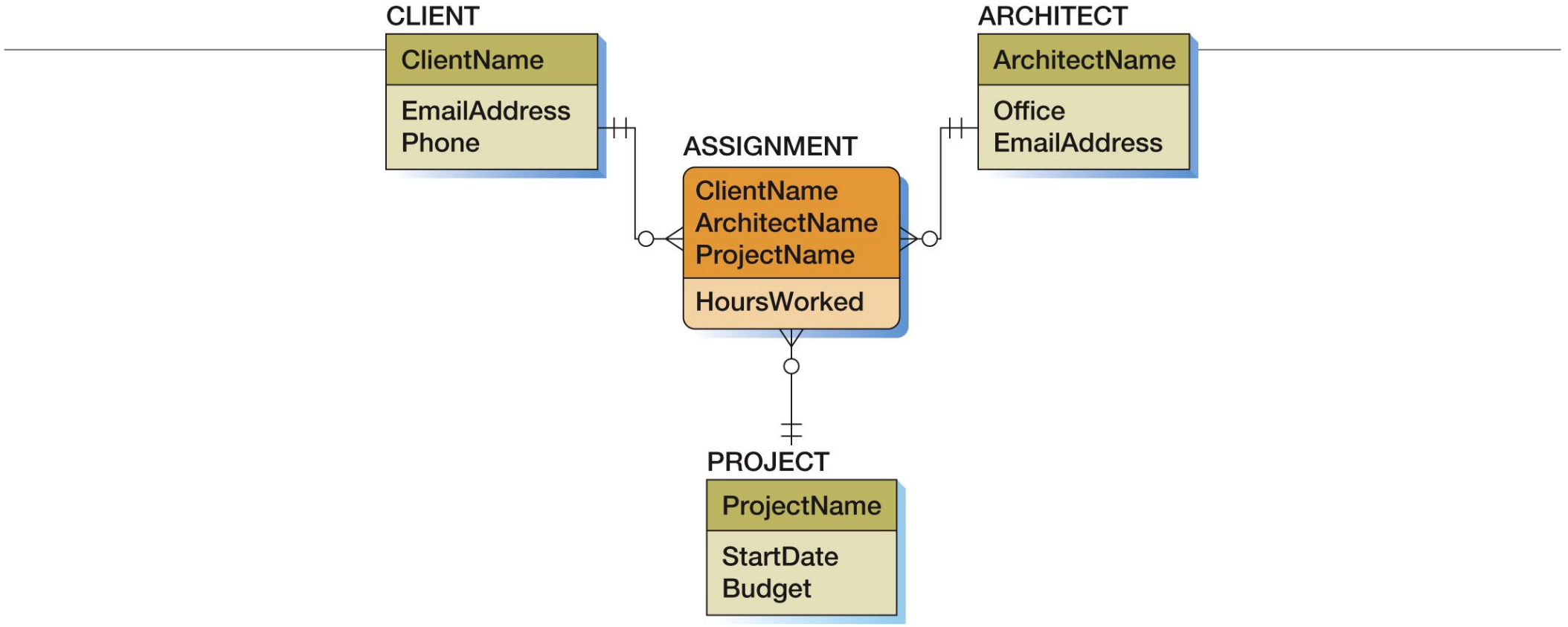


Figure 5-27 Form with Multivalued Attributes Fig. 5-26

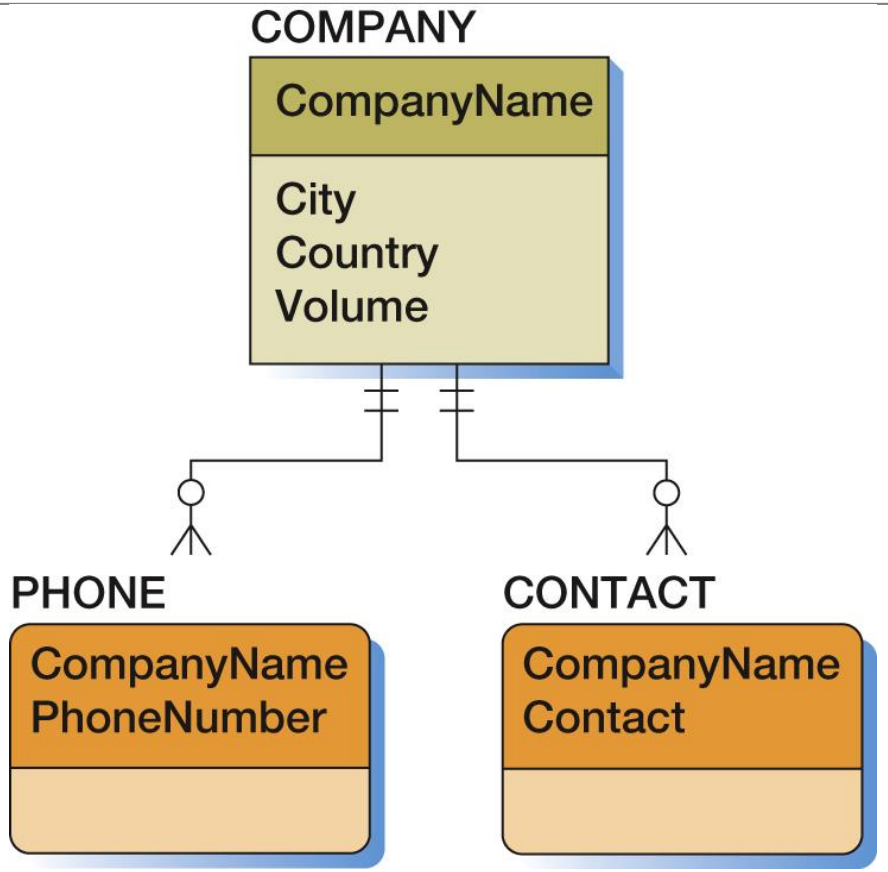


Figure 5-33 Sales Order in Figure 5-32

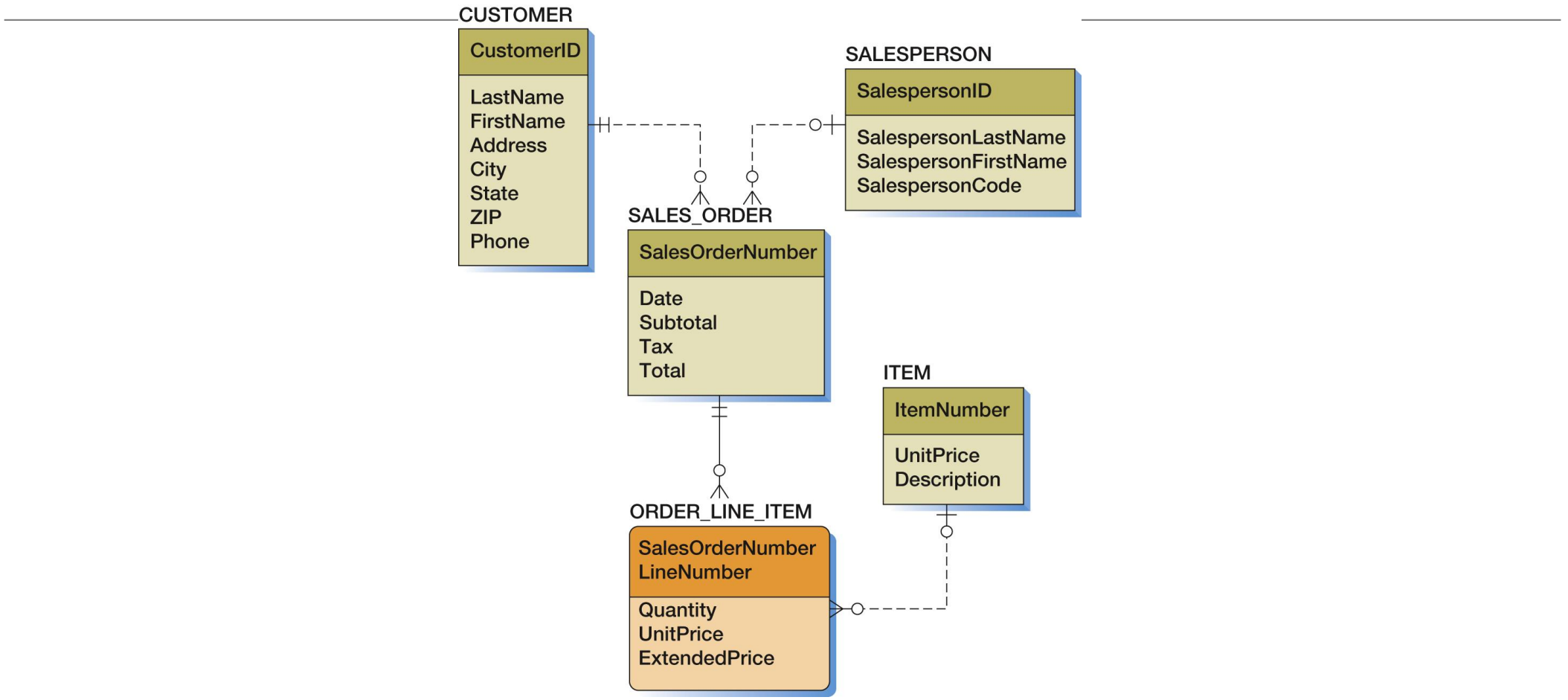


Figure 5-35: Employee Skills

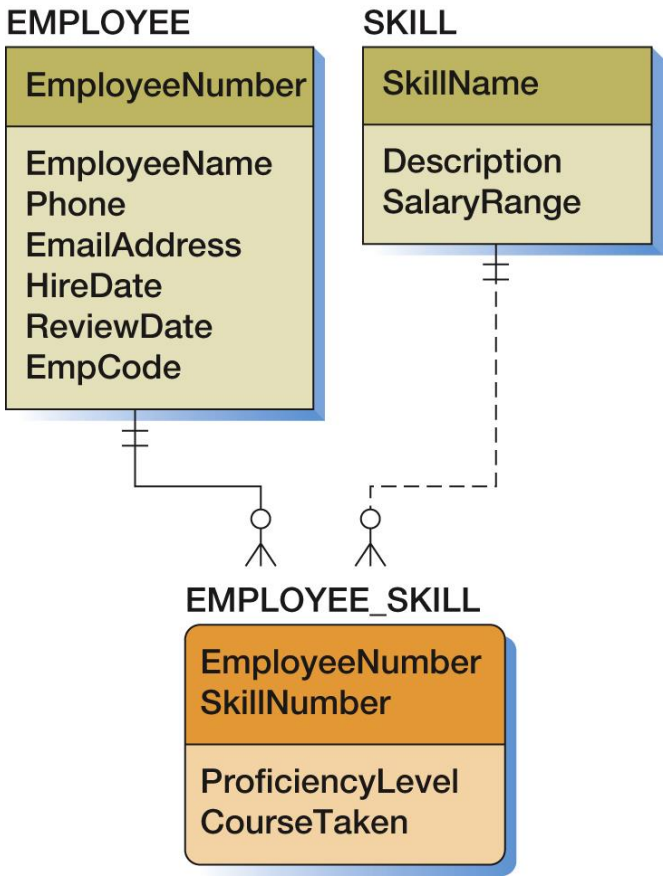
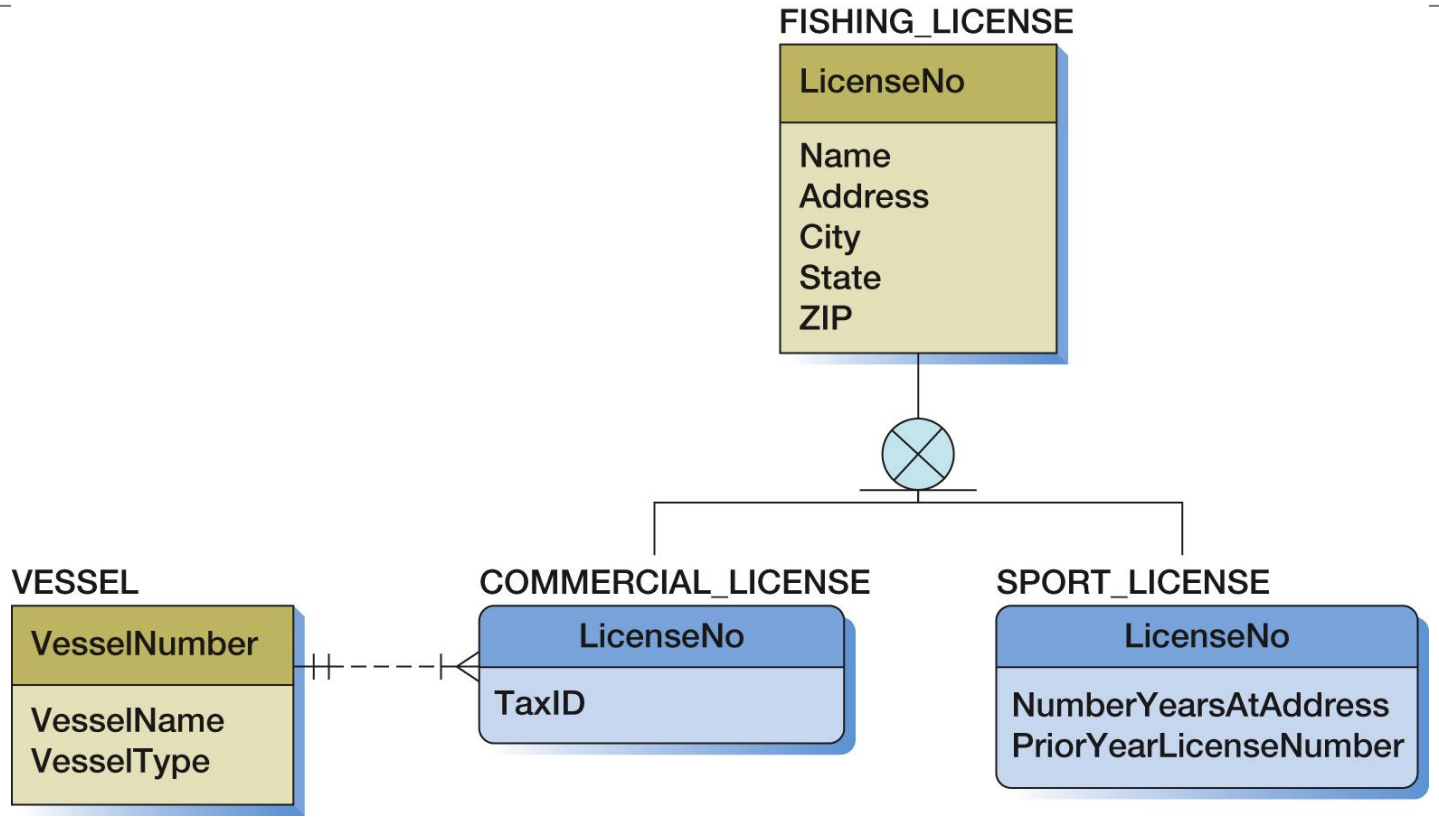


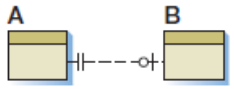
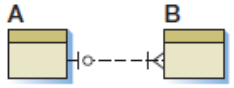
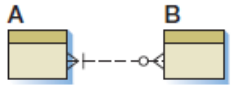
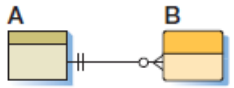
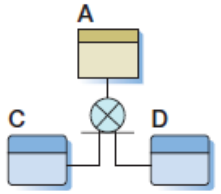
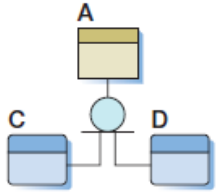
Figure 5-36 Data Entry Form Suggesting the Need for Subtypes

Resident Fishing License 2018 Season				**License No:** **03-1123432**	
Name:					
Street:					
City:		State:		ZIP:	
For Use by Commercial Fishers Only			For Use by Sport Fishers Only		
Vessel Number:			Number Years at This Address:		
Vessel Name:			Prior Year License Number:		
Vessel Type:					
Tax ID:					

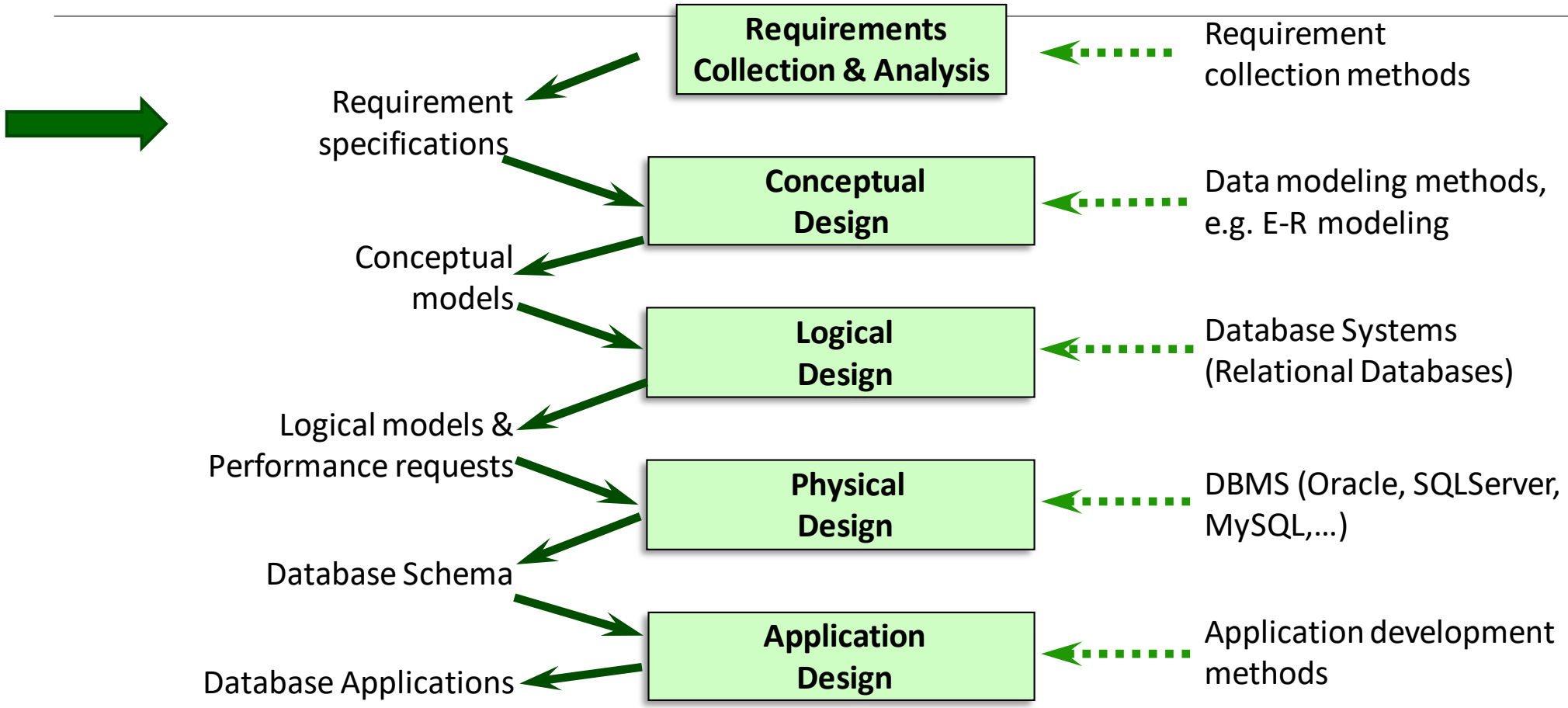
Figure 5-37 Data Model for Form in Figure 5-36



Crow's Feet Notation (Summary)

DEPARTMENT DepartmentName BudgetCode OfficeNumber	DEPARTMENT entity; DepartmentName is identifier; BudgetCode and OfficeNumber are attributes.
	1:1, nonidentifying relationship. A relates to zero or one B; B relates to exactly one A. Identifier and attributes not shown.
	1:N, nonidentifying relationship. A relates to one or many Bs; B relates to zero or one A. Identifier and attributes not shown.
	Many-to-many, nonidentifying relationship. A relates to zero or more Bs; B relates to one or more As. Identifier and attributes not shown.
	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 any combination of minimum and maximum cardinalities.
	A is supertype, C and D are exclusive subtypes. An entity may be a C or a D but not both. Discriminator not shown. Identifier and attributes not shown.
	A is supertype, C and D are inclusive subtypes. An entity may be a C or a D or both. Identifier and attributes not shown.

Recall: database design methodology



Conclusion

Data Model

- Model real world situation
  - [Potential Midterm Question: Why is this important?]
- Input from report and requirements (requirements part of database design)
- Entities (a thing of interest in the real world)
- Relationships (different types, importance of cardinality assignments)

Example: Highline University

Requirements analysis and conceptual modeling

Much to be studied and revisited. Note the similarities/differences between Chen notation and Crow's feet notation.