Chapter 5

Data Modeling with Entity-Relationship Diagram

Entities and Tables

- The principle difference between an entity and a table (relation) is that you can *express a relationship between entities without using foreign keys*.
- This makes it easier to work with entities in the early design process where the very existence of entities and the relationships between them is uncertain.

Chapter Objectives

- To understand the two-phase data *modeling*/database *design* process
- To understand the purpose of the data modeling process
- Logical/conceptual model vs. Physical model

Chapter Objectives

- To understand **entity-relationship (E-R) diagrams**
- To be able to determine **entities**, **attributes**, and **relationships**
- To be able to create **entity identifiers**
- To be able to determine **minimum and maximum cardinalities**
- To understand variations of the E-R model
- To understand and be able to use ID-dependent and other weak entities
- To understand and be able to use **supertype/subtype entities**

Chapter Objectives

- To understand and be able to use patterns

– strong entity patterns
– ID-dependent association pattern
– the ID-dependent multivalued attribute pattern
– the ID-dependent archetype/instance pattern
– the **line-item** pattern
– the **for-use-by** pattern
– the **recursive** pattern

- the *iterative* nature of the data modeling process
    - the data modeling process

Data modeling is one of critical success factors for an information system project.

1. THE DATA MODEL

Definition

- 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 then it is to change a database design, so it is the appropriate place to work through conceptual database problems.

E-R Model

- **Entity-Relationship model** is a set of concepts and graphical symbols that can be used to create conceptual schemas.
- History of the E-R model
  - **Original E-R model**—by Peter Chen (1976)
  - **Extended E-R model (subtypes)**—extensions to the Chen model
  - **Information Engineering (IE)**—by James Martin (1990); it uses "crow's foot" notation, is easier to understand.
  - **IDEF1X**—a national standard developed by the National Institute of Standards and Technology [see *Appendix C*]
  - **Unified Modeling Language (UML)**—by the Object Management Group; it supports object-oriented methodology [see *Appendix D*]

Entities

- Something that can be identified and the users want to track:
  - **Entity class**—a collection of entities of a given type
  - **Entity instance**—the occurrence of a particular entity
- There are usually many instances of an entity in an entity class.

Example
The Entity Class CUSTOMER and Two Entity Instances

CUSTOMER Entity

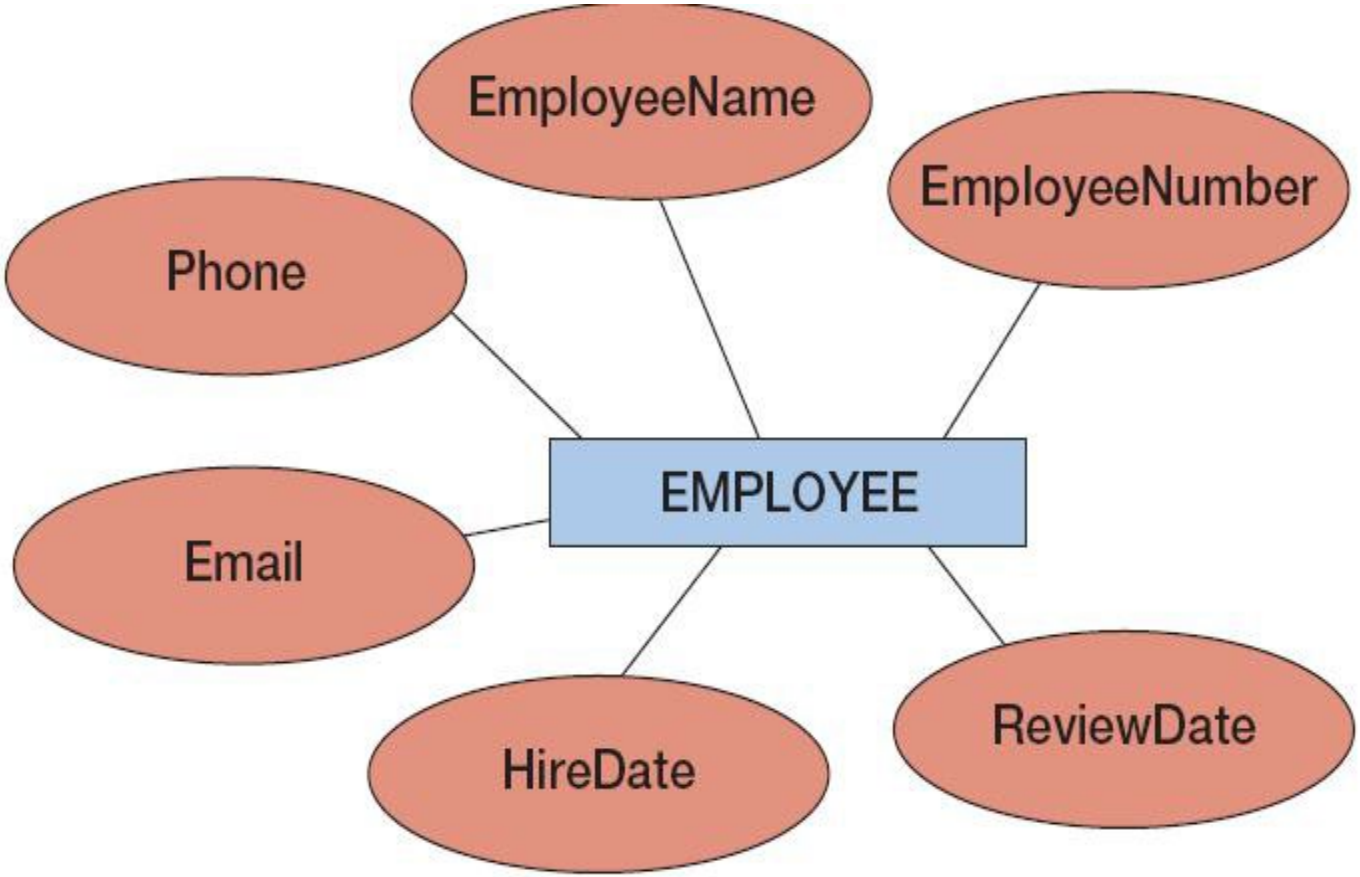
Two CUSTOMER Instances

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

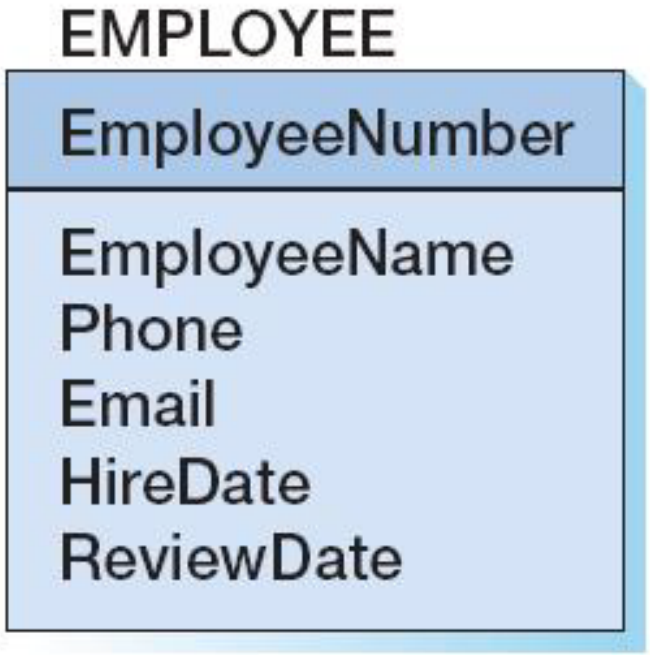
Attributes

- **Attributes** describe an entity's characteristics.
  - All entity instances of a given entity class have the same attributes, but vary in the values of those attributes.
- Originally shown in data models as **ellipses**.
- Data modeling products today commonly show attributes in **rectangular form**.

Example
The EMPLOYEE Entity and its Attributes in Ellipses



Example
The EMPLOYEE Entity and its Attributes in a Box



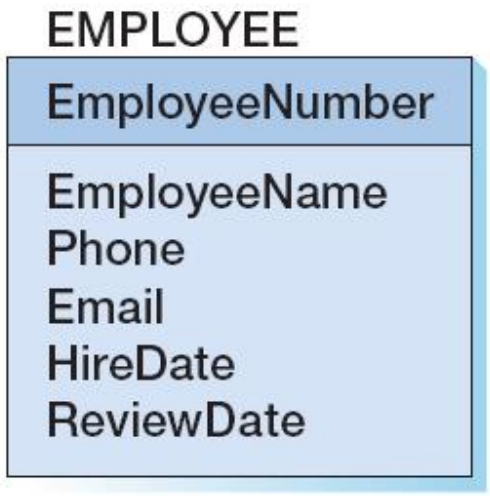
Identifiers

- **Identifiers** are attributes that name, or identify, entity instances.
- The identifier of an entity instance consists of one or more attributes.
- **Composite identifiers** are identifiers that consist of two or more attributes.
- Identifiers in data models become keys in database designs.
  - Entities have identifiers.
  - Tables (or relations) have keys.

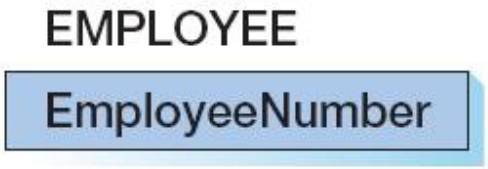
Weak Entities

- A weak entity is an entity whose existence depends upon another entity.
- All *ID-Dependent* entities are considered weak.
- Example: Employees and their dependents

Entity-Attribute Display in Data Models



(a) Entity with All Attributes



(b) Entity with Identifier Attribute Only

(c) Entity with No Attributes

Relationships

- Entities can be associated with one another in **relationships**:
  - **Relationship classes**: associations among entity classes
  - **Relationship instances**: associations among entity instances

- *In the original E-R model, relationships could have attributes.*

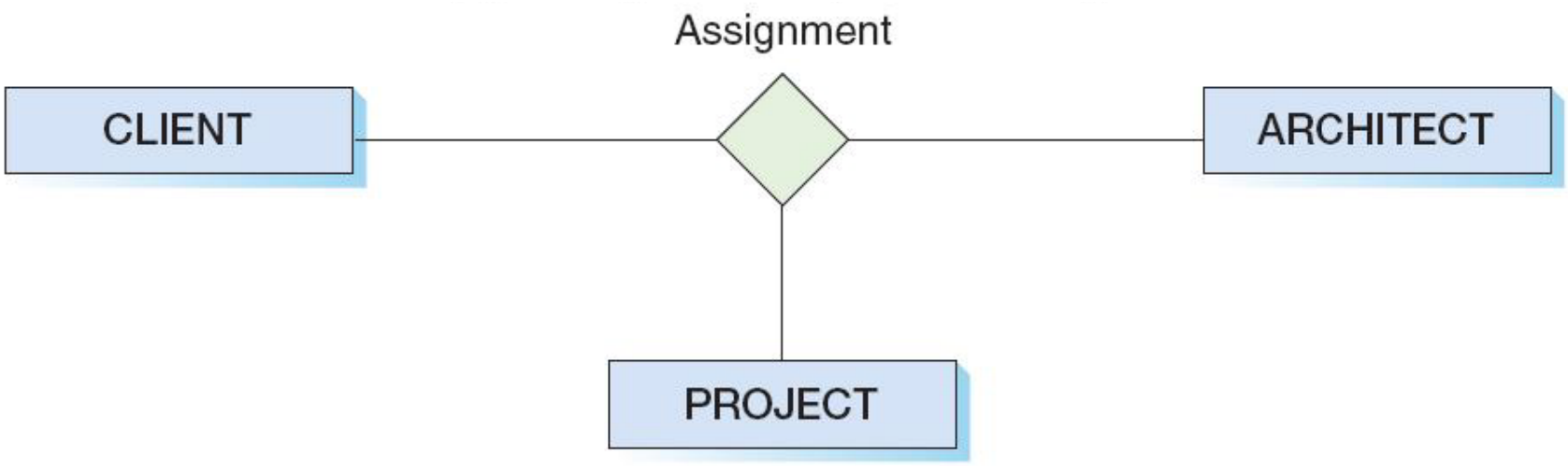
- A relationship class can involve two or more

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.

Binary Relationship

Ternary Relationship



How many instances can/must participate in a relationship?

2. CARDINALITY (MULTIPLICITY)

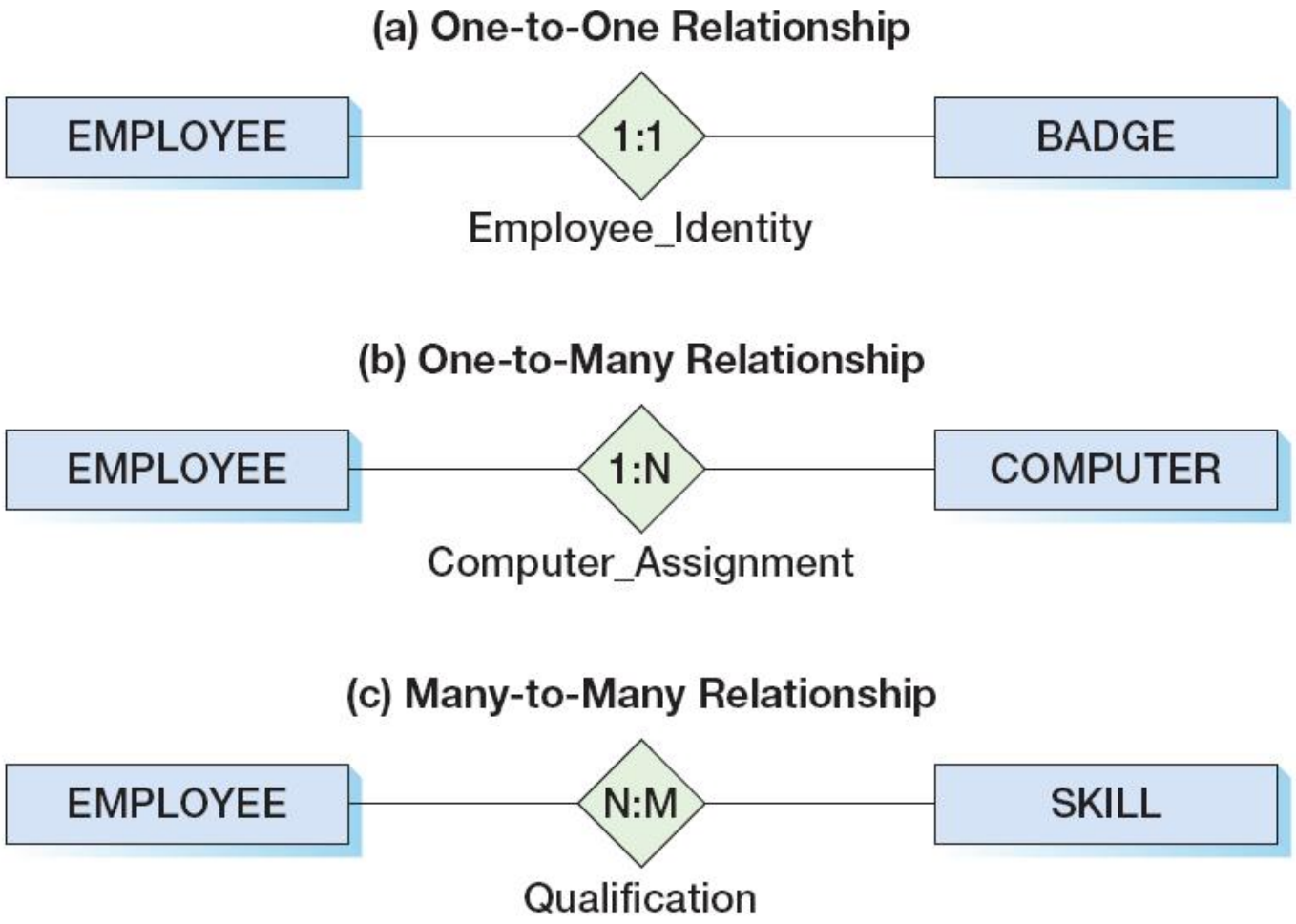
Cardinality / Multiplicity

- **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.

Maximum Cardinality

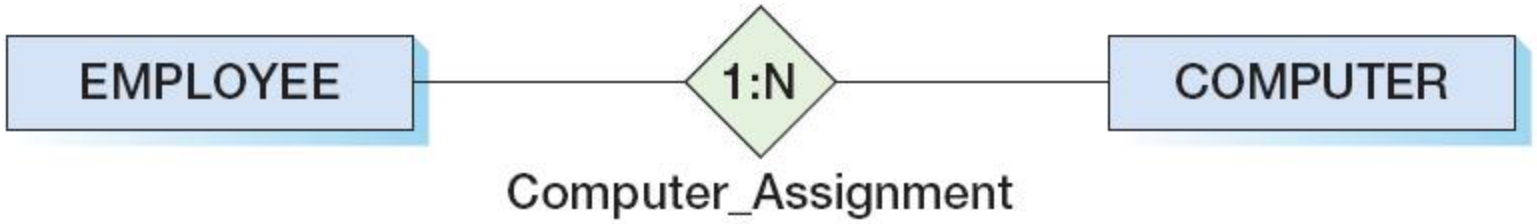
- **Maximum cardinality** is the maximum number of entity instances that ***can*** participate in a relationship.
- There are three types of maximum cardinality:
  - One-to-One [1:1]
  - One-to-Many [1:N]
  - Many-to-Many [N:M]

Three Types of Maximum Cardinality



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:



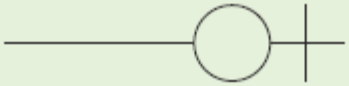
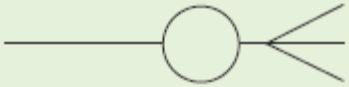
HAS-A Relationships

- 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 one*** assigned EMPLOYEE.

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.

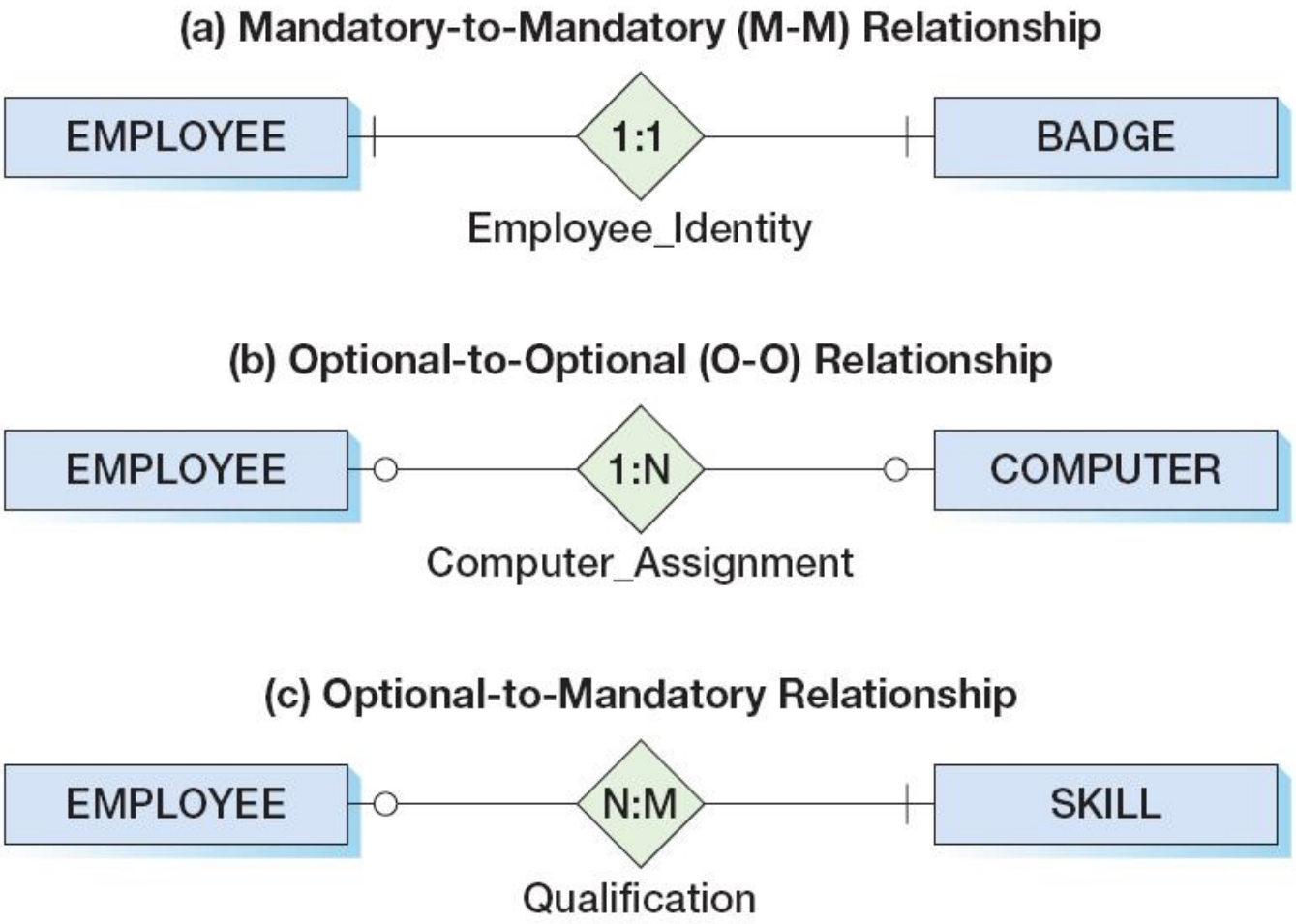
Data Modeling Notation- IE Crow's Foot I

Symbol	Meaning
	Mandatory—One
	Mandatory—Many
	Optional—One
	Optional—Many

Reading Minimum Cardinality

- Look toward the entity in question:
  - 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]).

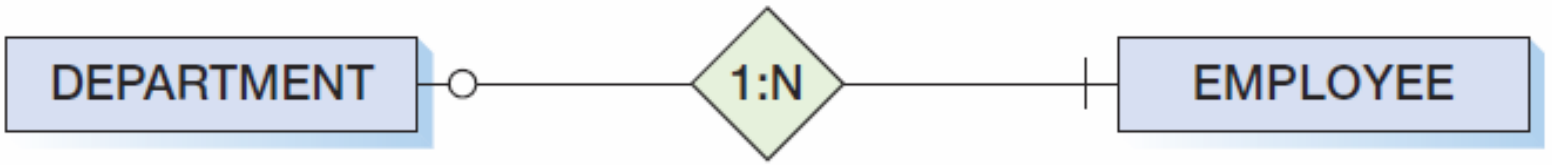
Three Types of Minimum Cardinality



Diagramming your data model

3. DATA MODELING NOTATIONS

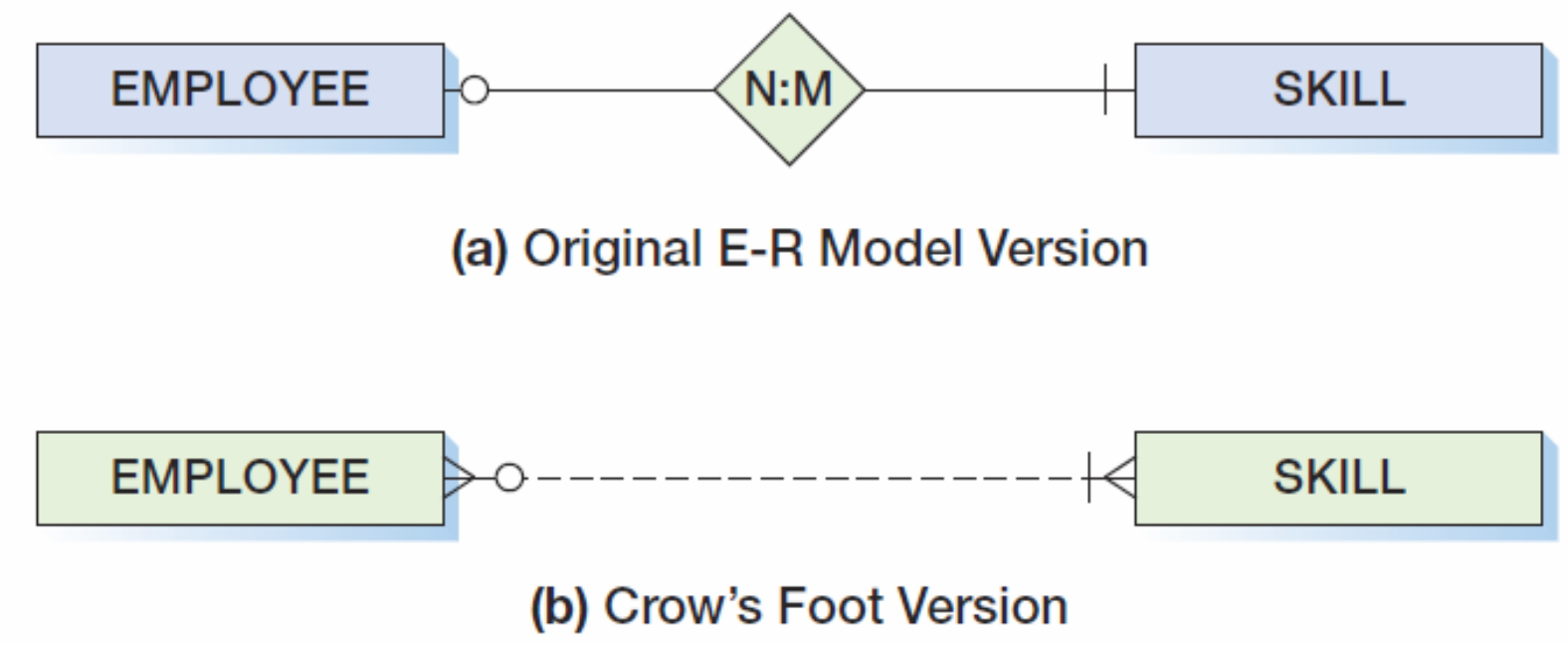
Data Modeling Notation- IE Crow's Foot 1:N



(a) Original E-R Model Version

(b) Crow's Foot Version

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

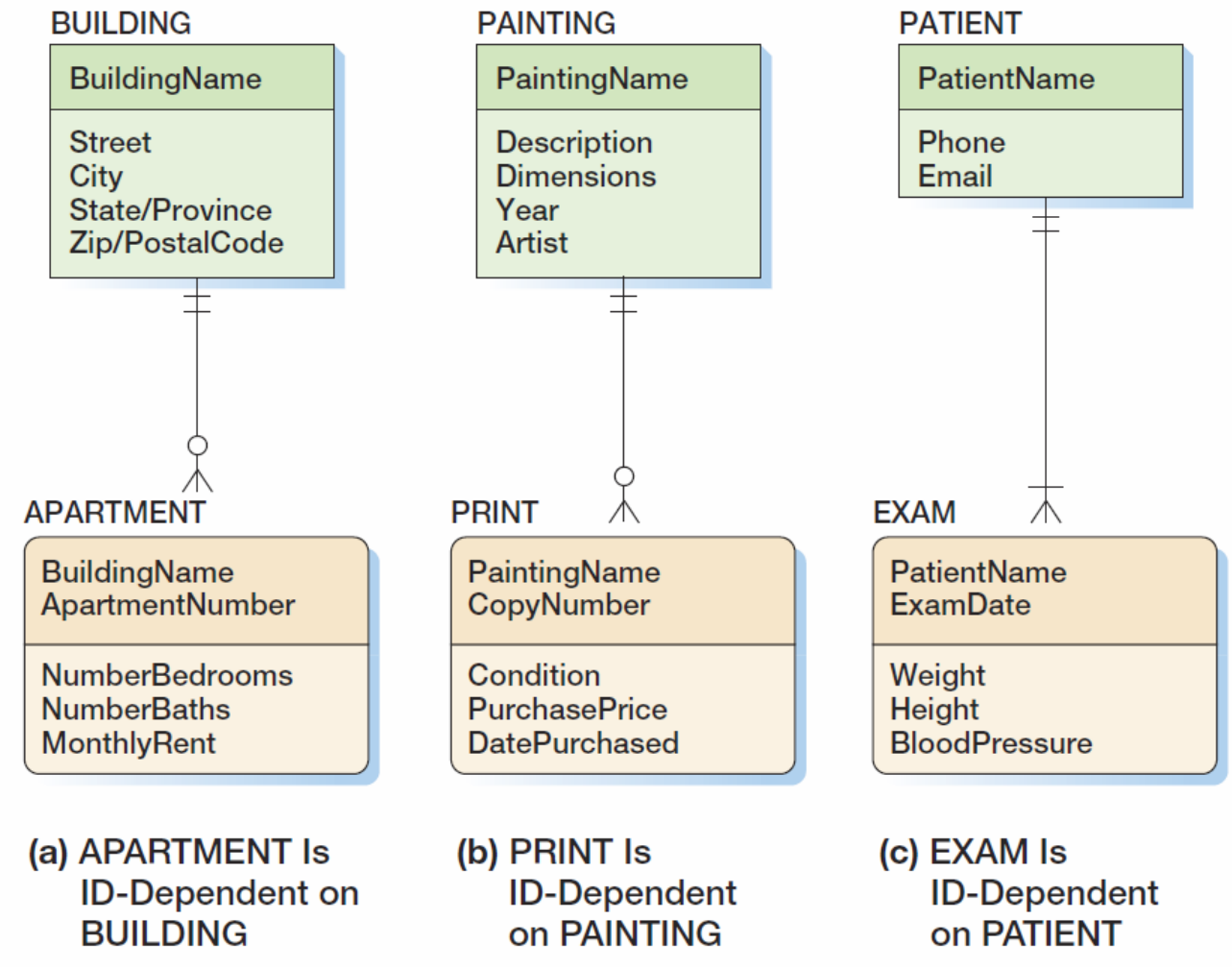


4. TYPES OF ENTITIES

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*.

ID-Dependent Entities

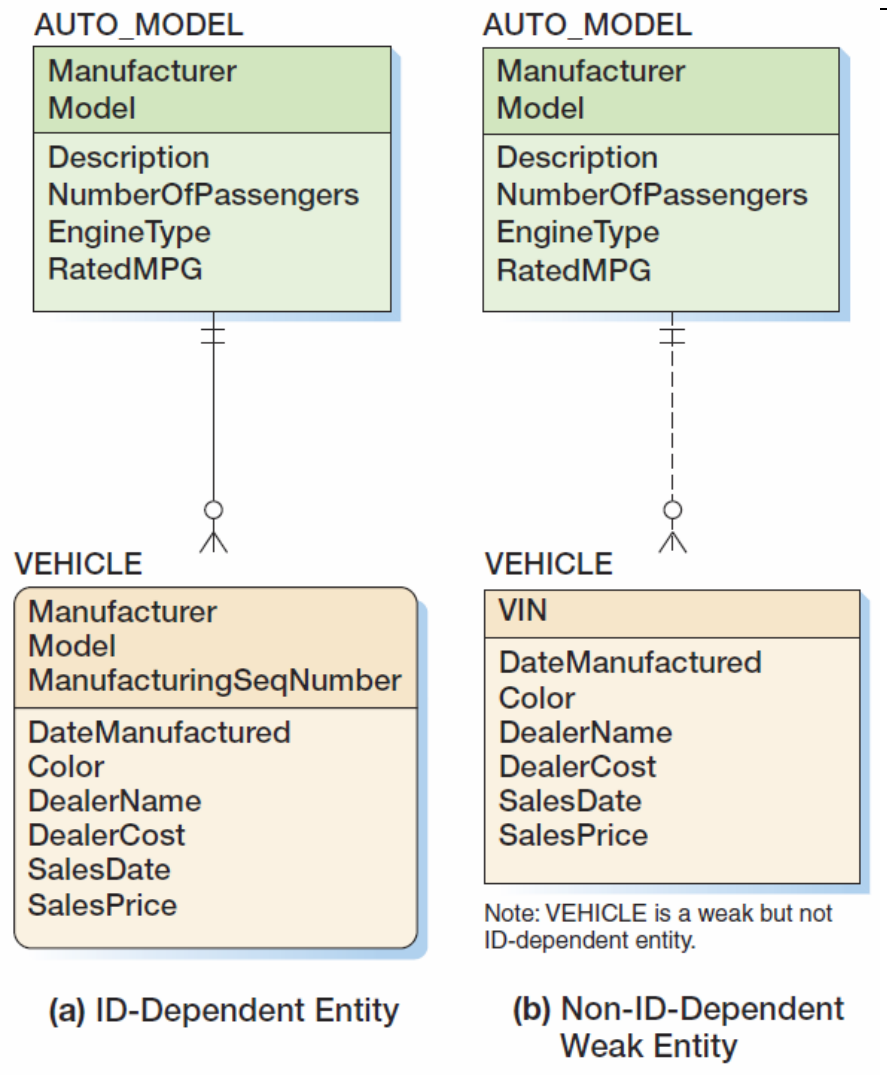


(a) APARTMENT Is ID-Dependent on BUILDING

(b) PRINT Is ID-Dependent on PAINTING

(c) EXAM Is ID-Dependent on PATIENT

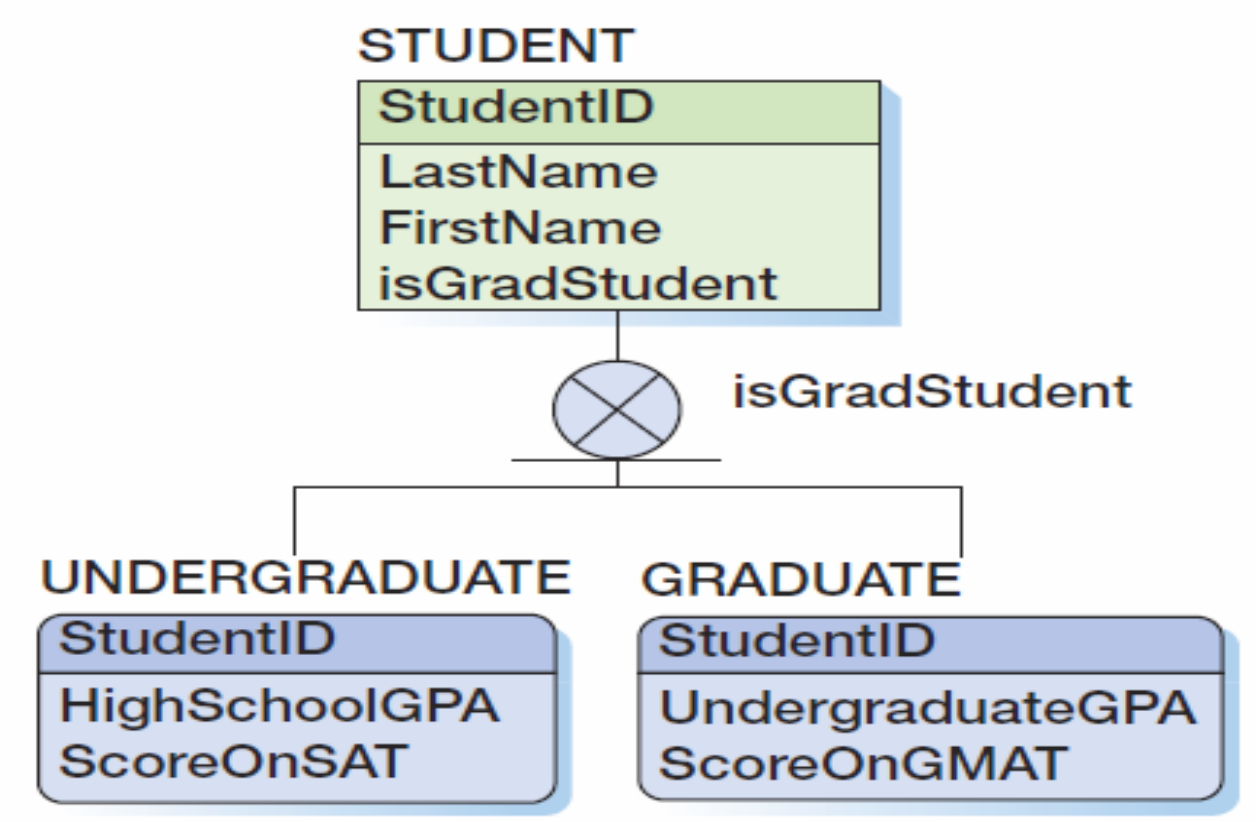
Weak Entities



Subtype Entities

- A **subtype entity** is a special case of a **supertype entity**:
  - STUDENT : UNDERGRADATE or GRADUATE
- The *supertype* contains all common attributes, while the subtypes contain specific attributes.
- The *supertype* may have a **discriminator** attribute which indicates the subtype

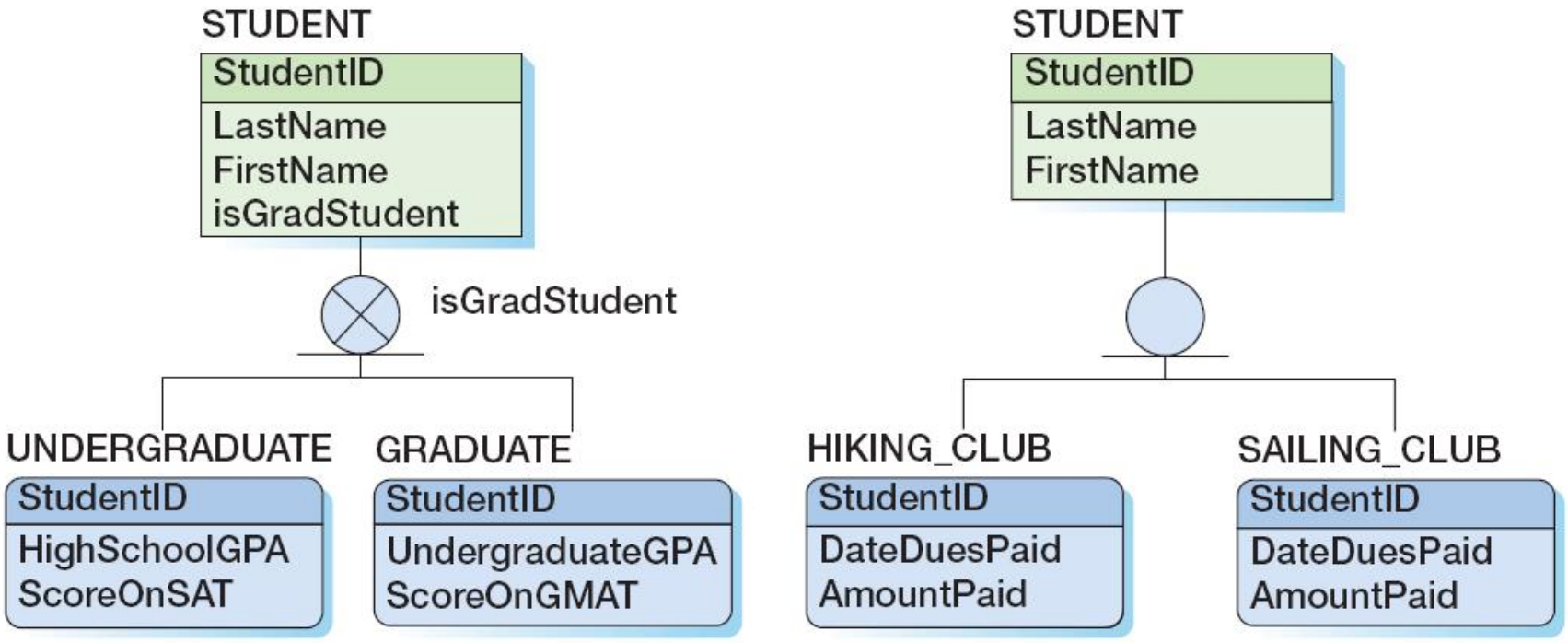
Subtypes with a Discriminator



Subtypes: Exclusive or Inclusive

- If subtypes are **exclusive**, one supertype relates to at most one subtype.
- If subtypes are **inclusive**, one supertype can relate to one or more subtypes.

Subtypes: Exclusive or Inclusive



(a) Exclusive Subtypes with Discriminator

(b) Inclusive Subtypes

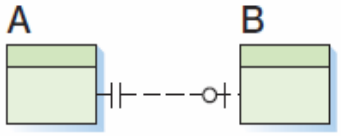
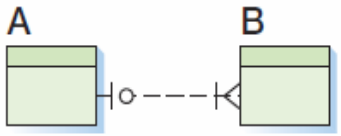
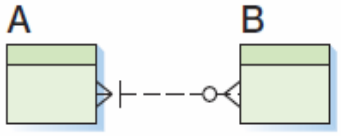
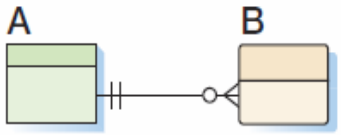
Subtypes: IS-A Relationships

- Relationships connecting supertypes and subtypes are called **IS-A relationships**, because a subtype ***is a*** supertype.
- The identifier 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).
- Subtypes are used to avoid **value-inappropriate nulls**.

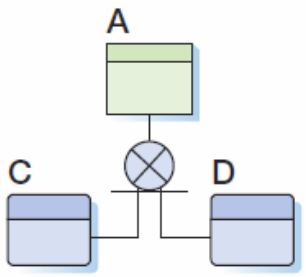
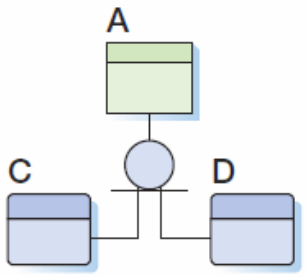
You will see these symbols in many of database diagrams.

5. CROW'S FOOT SYMBOL

IE Crow's Foot Symbol Summary I

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.
	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.

IE Crow's Foot Symbol Summary II

	A is supertype, C and D are exclusive subtypes. Discriminator not shown. Identifier and attributes not shown.
	A is supertype, C and D are inclusive subtypes. Identifier and attributes not shown.

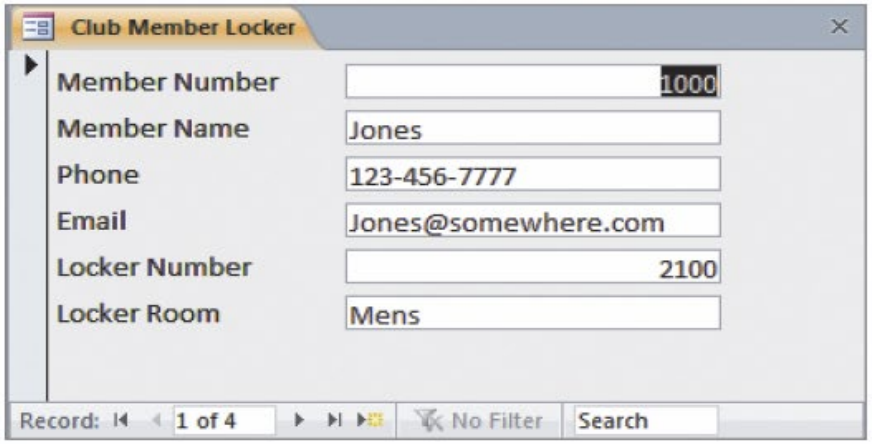
Common patterns existing among entities

6. DESIGN PATTERNS

Strong Entity Patterns

Strong Entity Patterns-
1:1 Strong Entity Relationships

(a) Club Membership Data Entry Form

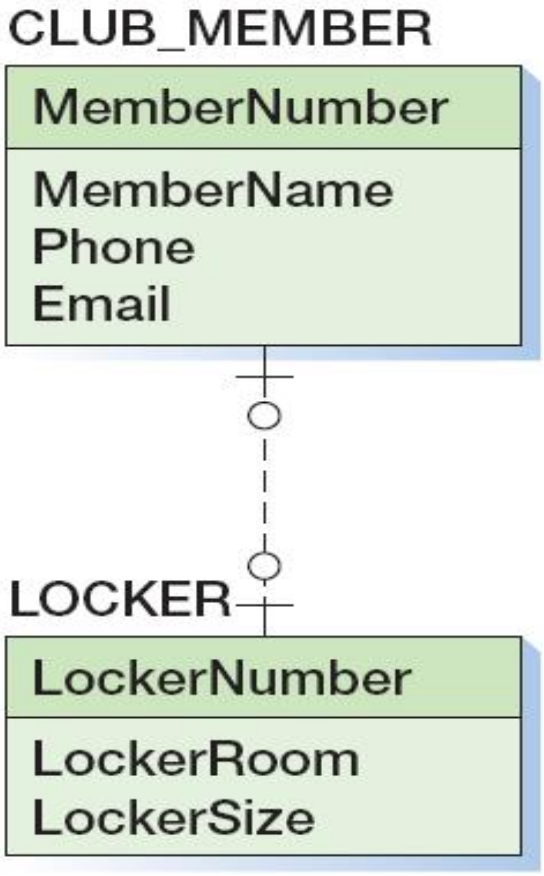


(b) Club Locker Report

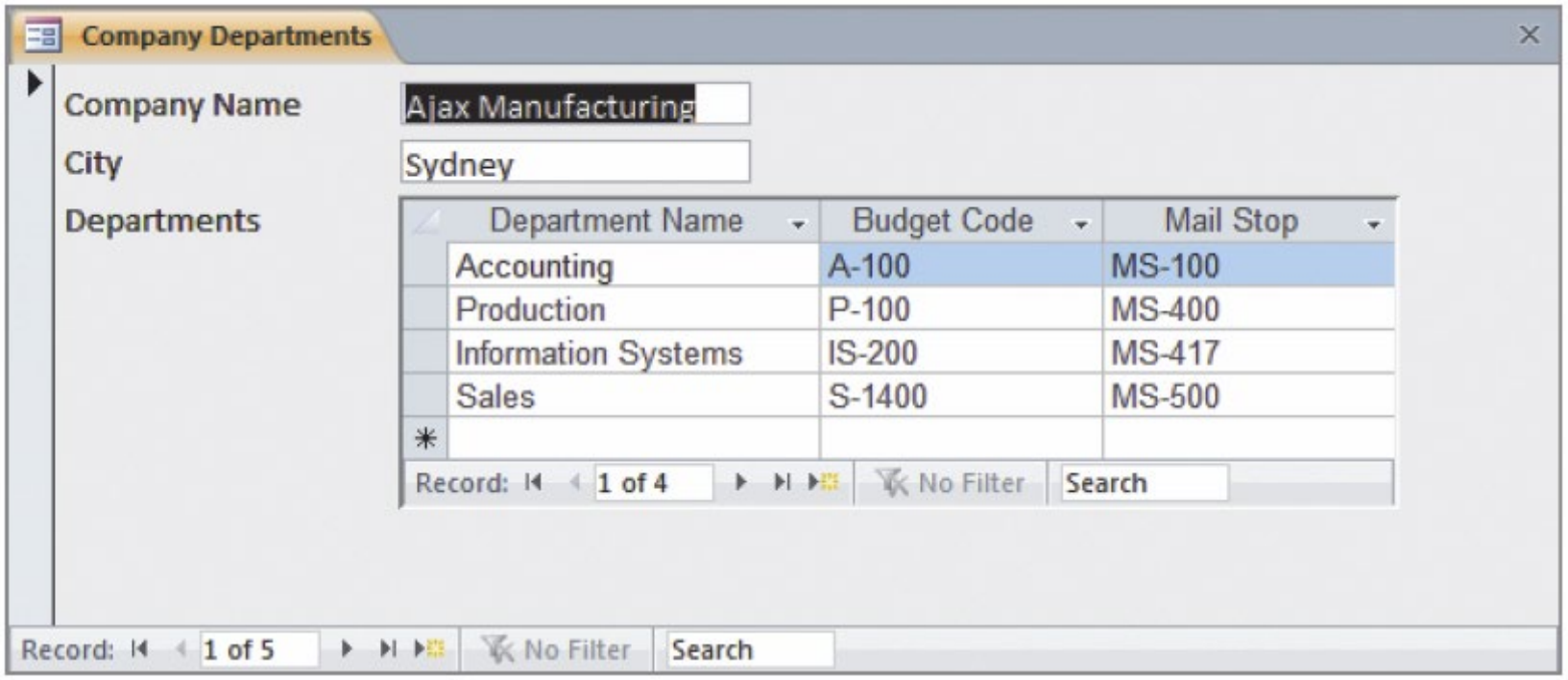
CLUB LOCKERS

Member Number	Member Name	Locker Number	Locker Room	Locker Size
1000	Jones	2100	Mens	Med
2000	Abernathy	2200	Womens	Large
3000	Wu	2115	Mens	Large
4000	Lai	2217	Womens	Small

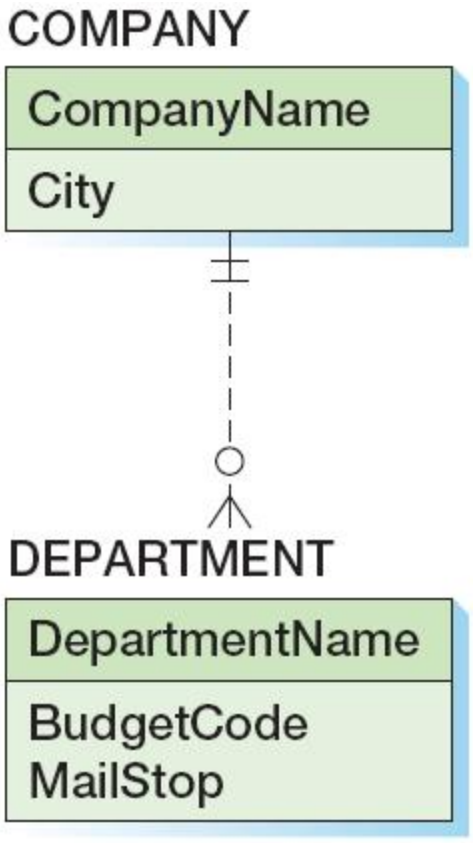
Strong Entity Patterns-
1:1 Strong Entity Relationships



Strong Entity Patterns: 1:N Strong Entity Relationships

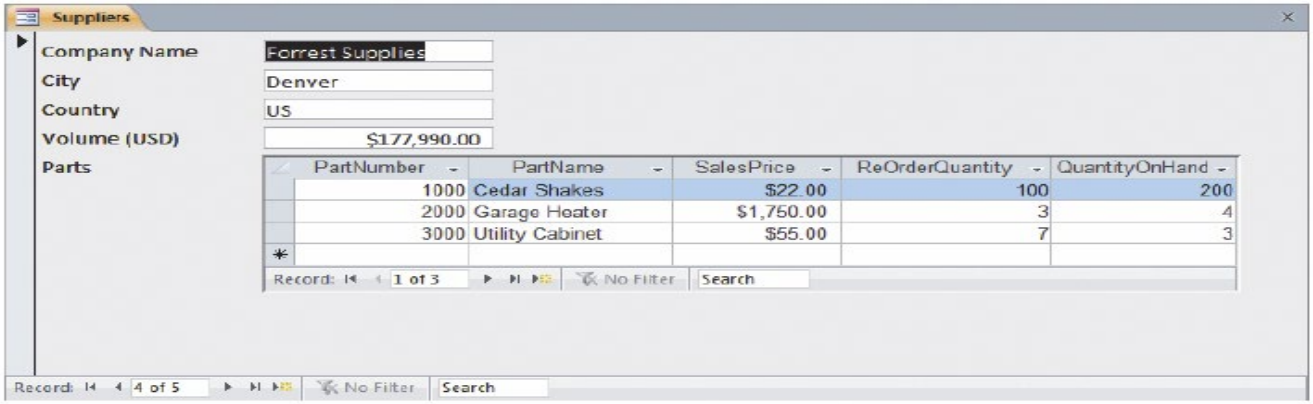


Strong Entity Patterns: 1:N Strong Entity Relationships



Strong Entity Patterns: N:M Strong Entity Relationships

(a) SUPPLIERS Form

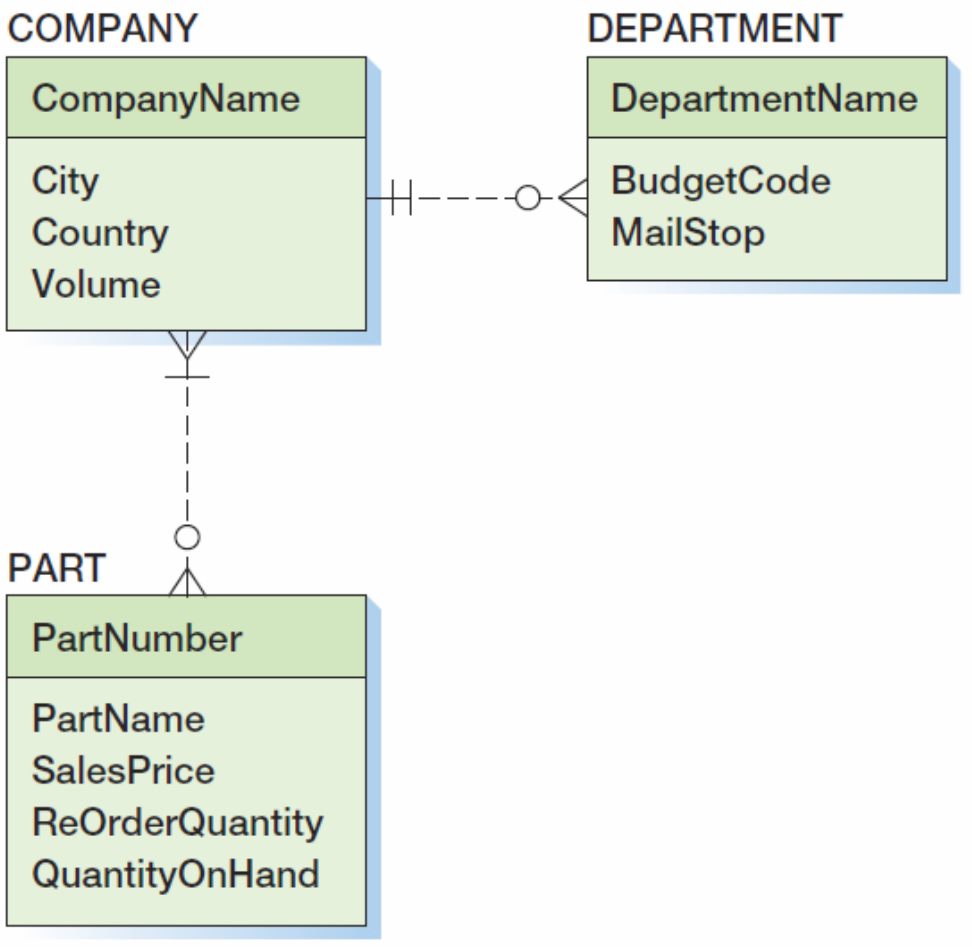


(b) PART Report

PART

Part Number	Part Name	Sales Price	ROQ	QOH	Company Name	City	Country
1000	Cedar Shakes	$22.00	100	200			
					Bristol Systems	Manchester	England
					ERS Systems	Vancouver	Canada
					Forrest Supplies	Denver	US
2000	Garage Heater	$1,750.00	3	4			
					Bristol Systems	Manchester	England
					ERS Systems	Vancouver	Canada
					Forrest Supplies	Denver	US
					Kyoto Importers	Kyoto	Japan
3000	Utility Cabinet	$55.00	7	3			
					Ajax Manufacturing	Sydney	Australia
					Forrest Supplies	Denver	US

Strong Entity Patterns: N:M Strong Entity Relationships



ID-Dependent Patterns

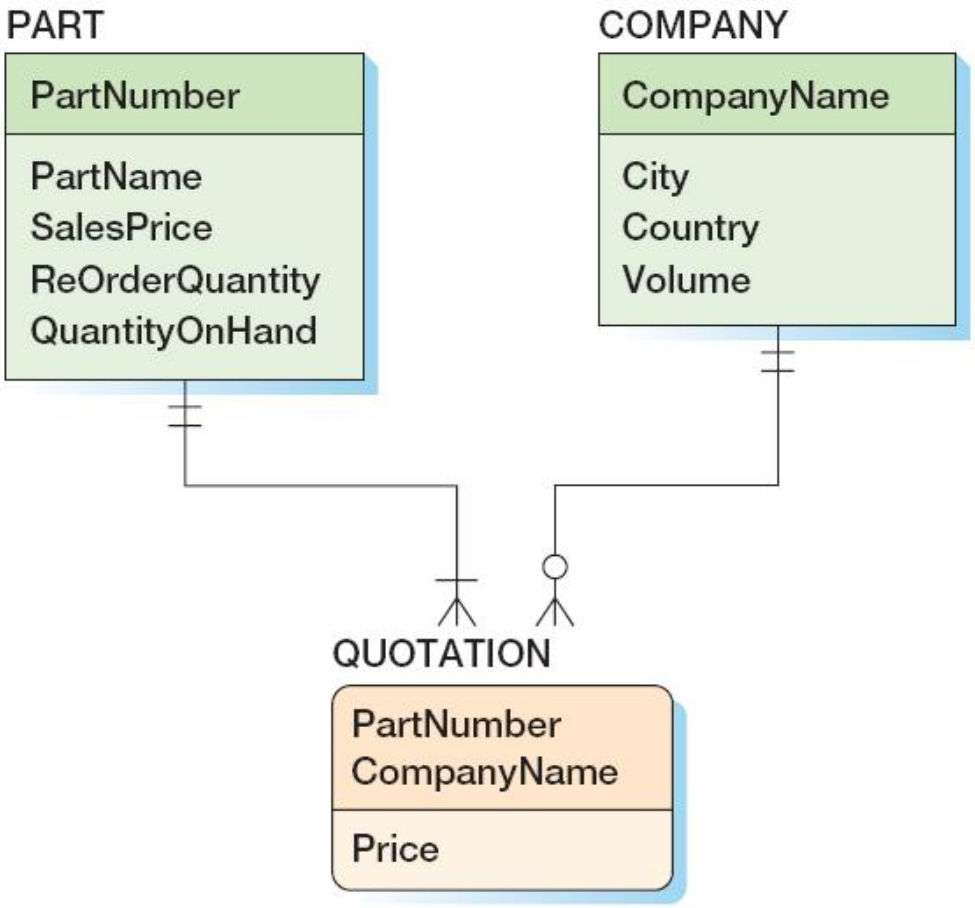
- The Association Pattern
- The Multivalued Attribute Pattern
- The Composite Multivalued Attributes Pattern
- The Archetype/Instance Pattern

ID-Dependent Relationships:
The Association Pattern

PART QUOTATIONS

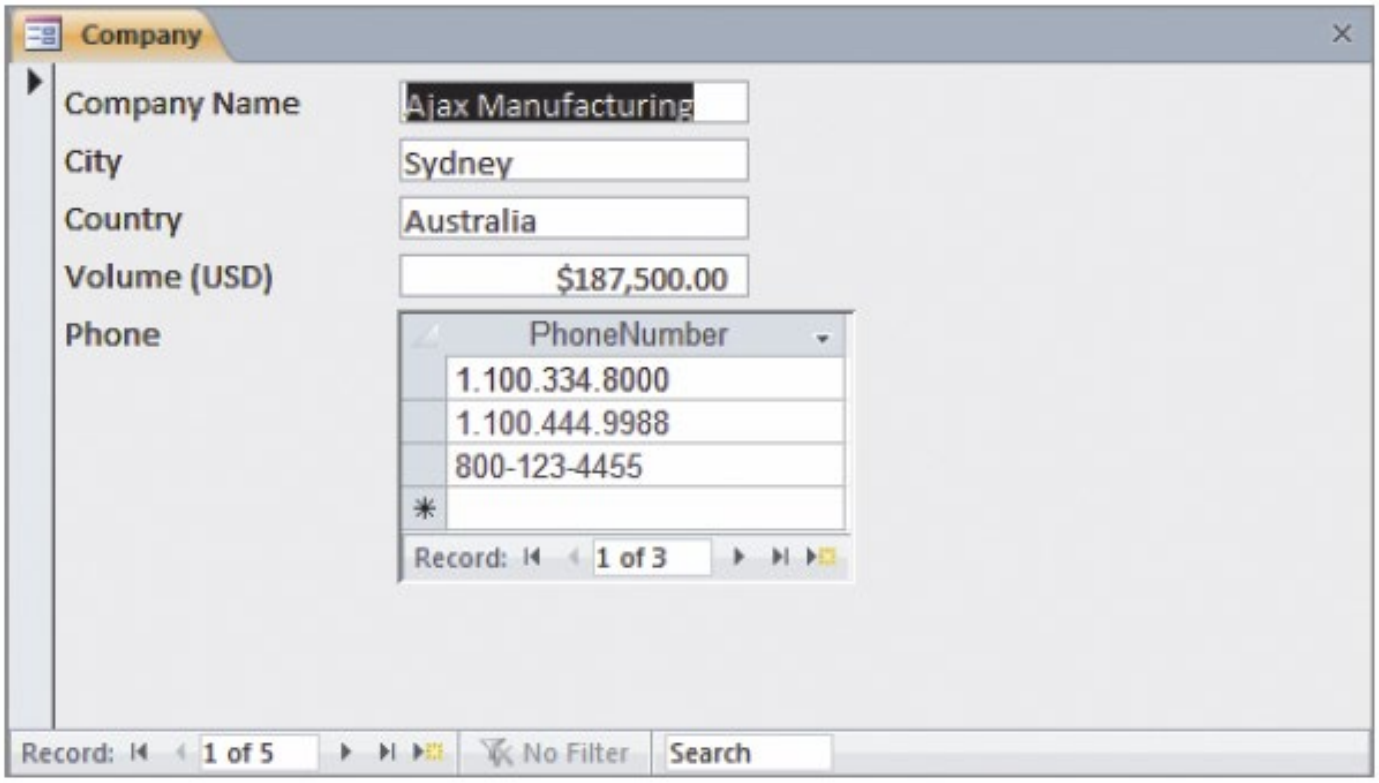
PartNumber	PartName	SalesPrice	ROQ	QOH	CompanyName	City	Country	Price
1000	Cedar Shakes	$22.00	100	200				
					Bristol Systems	Manchester	England	$14.00
					ERS Systems	Vancouver	Canada	$12.50
					Forrest Supplies	Denver	US	$15.50
2000	Garage Heater	$1,750.00	3	4				
					Bristol Systems	Manchester	England	$950.00
					ERS Systems	Vancouver	Canada	$875.00
					Forrest Supplies	Denver	US	$915.00
					Kyoto Importers	Kyoto	Japan	$1,100.00
3000	Utility Cabinet	$55.00	7	3				
					Ajax Manufacturing	Sydney	Australia	$37.50
					Forrest Supplies	Denver	US	$42.50

ID-Dependent Relationships: The Association Pattern

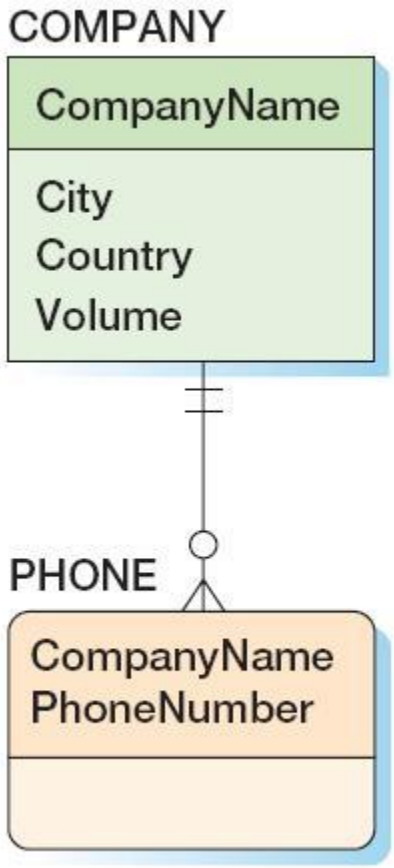


In database design, a table pattern where an intersection table contains additional attributes beyond the attributes that make up the composite primary key.

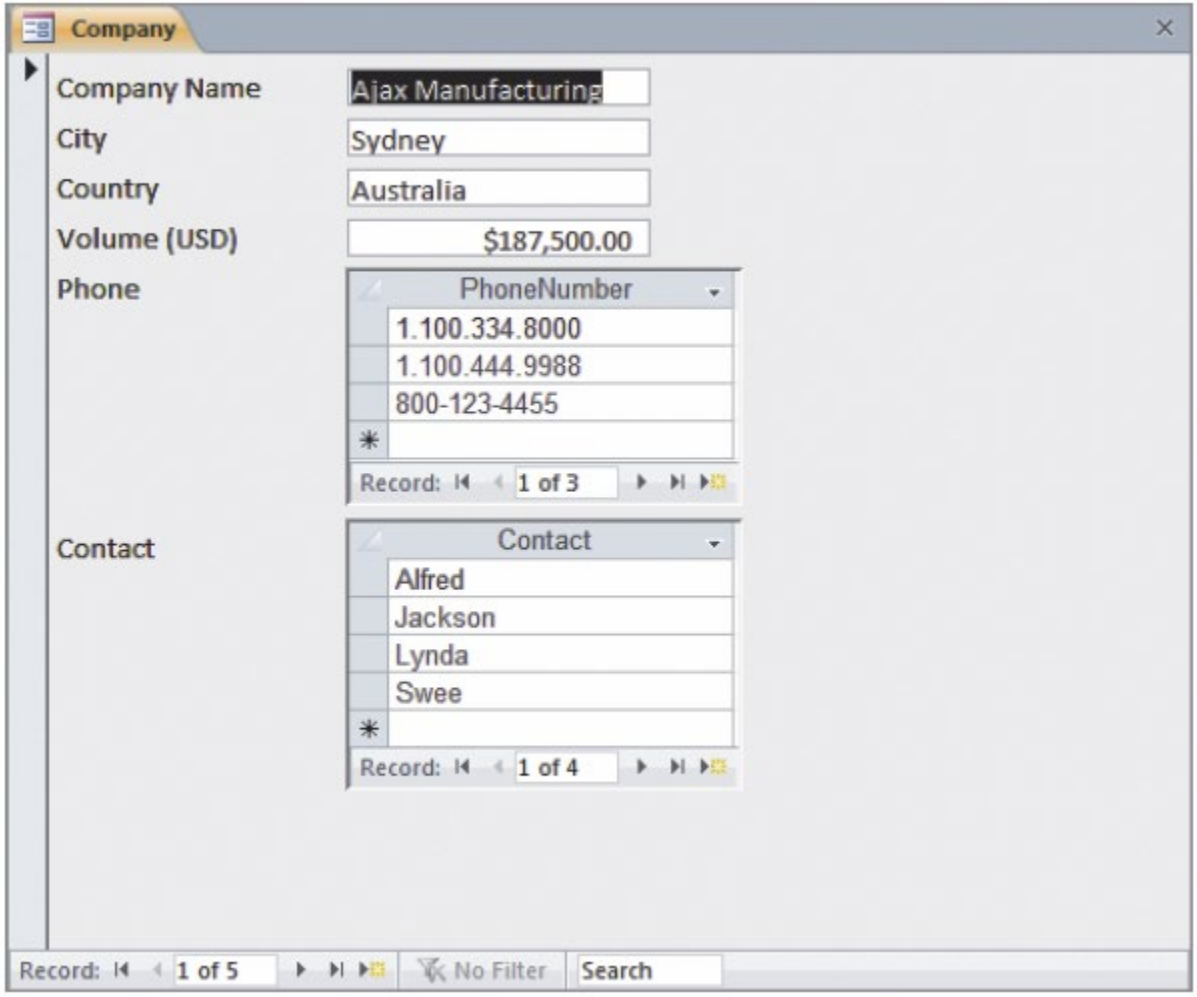
ID-Dependent Relationships:
The Multivalued Attribute Pattern



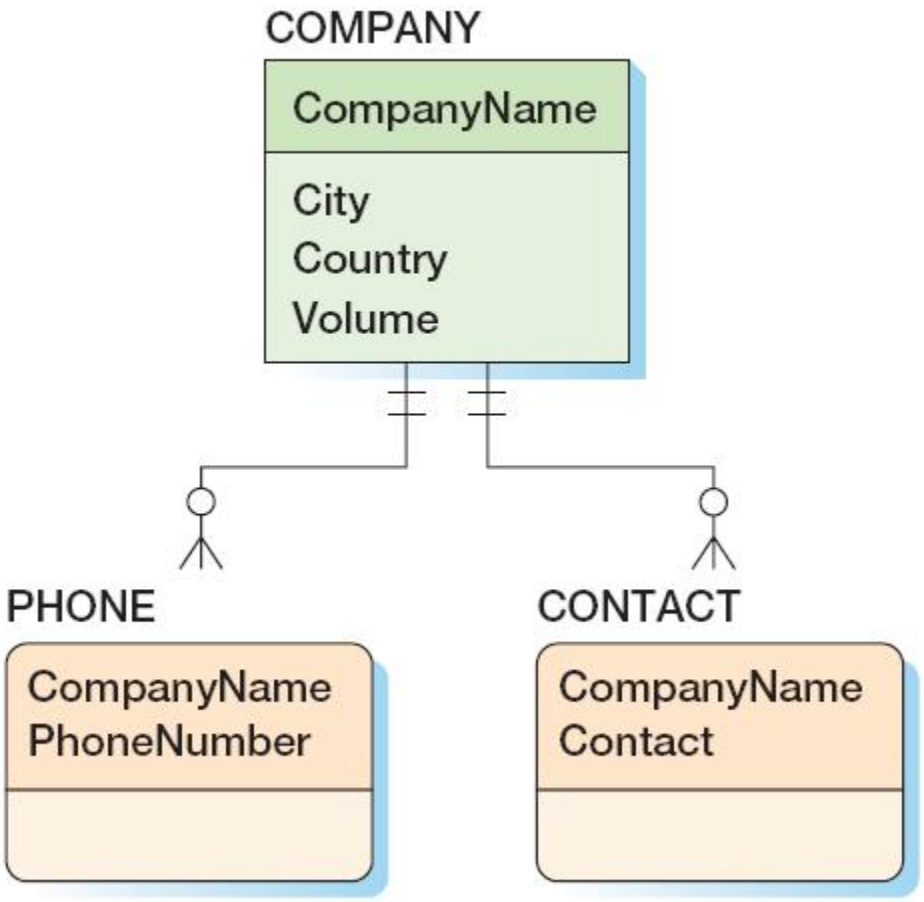
ID-Dependent Relationships: The Multivalued Attribute Pattern



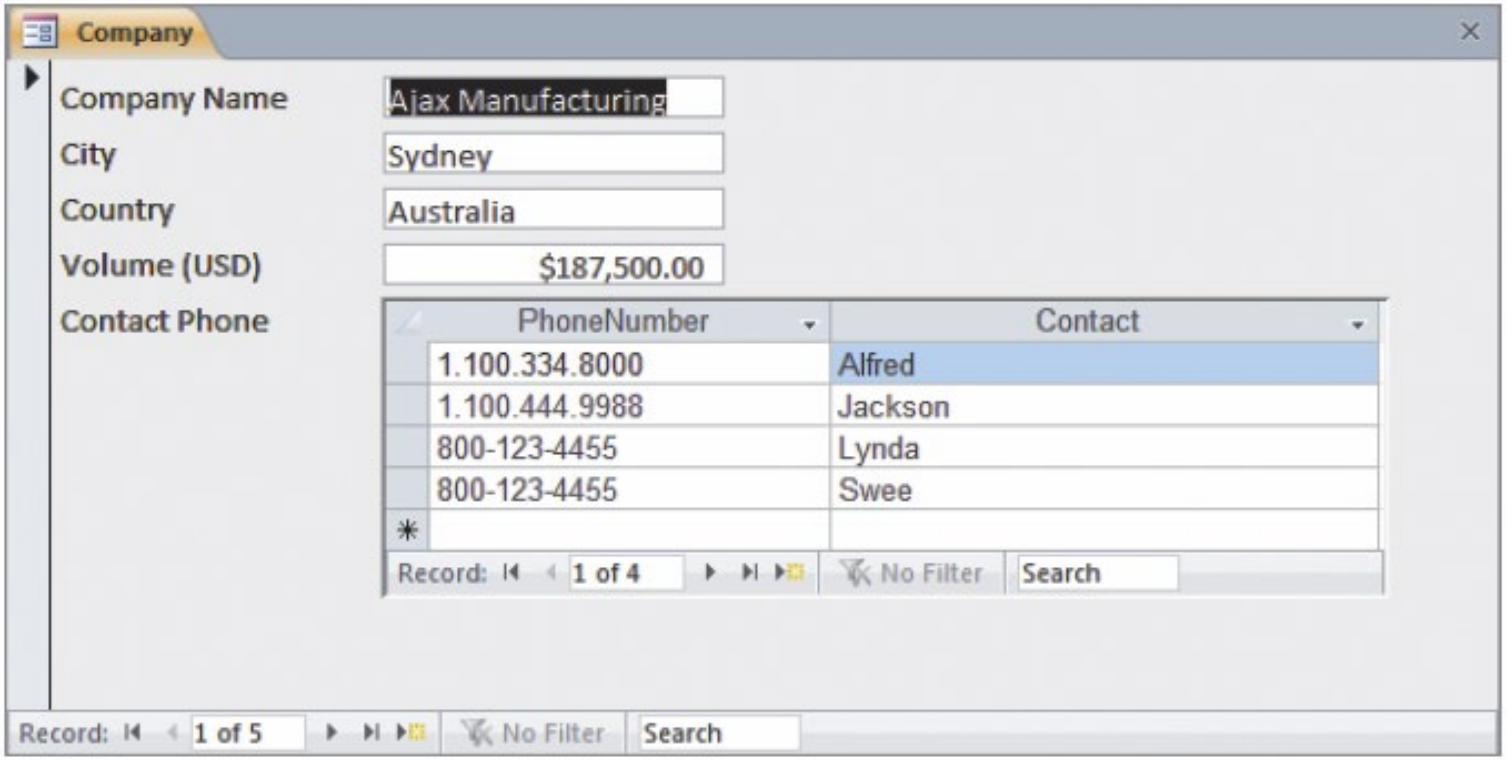
ID-Dependent Relationships: The Multivalued Attribute Pattern



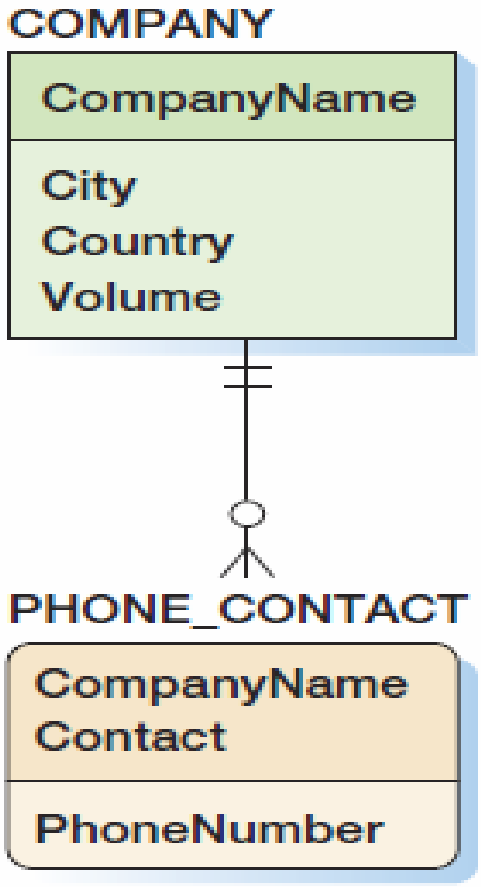
ID-Dependent Relationships: The Multivalued Attribute Pattern



ID-Dependent Relationships: Composite Multivalued Attributes



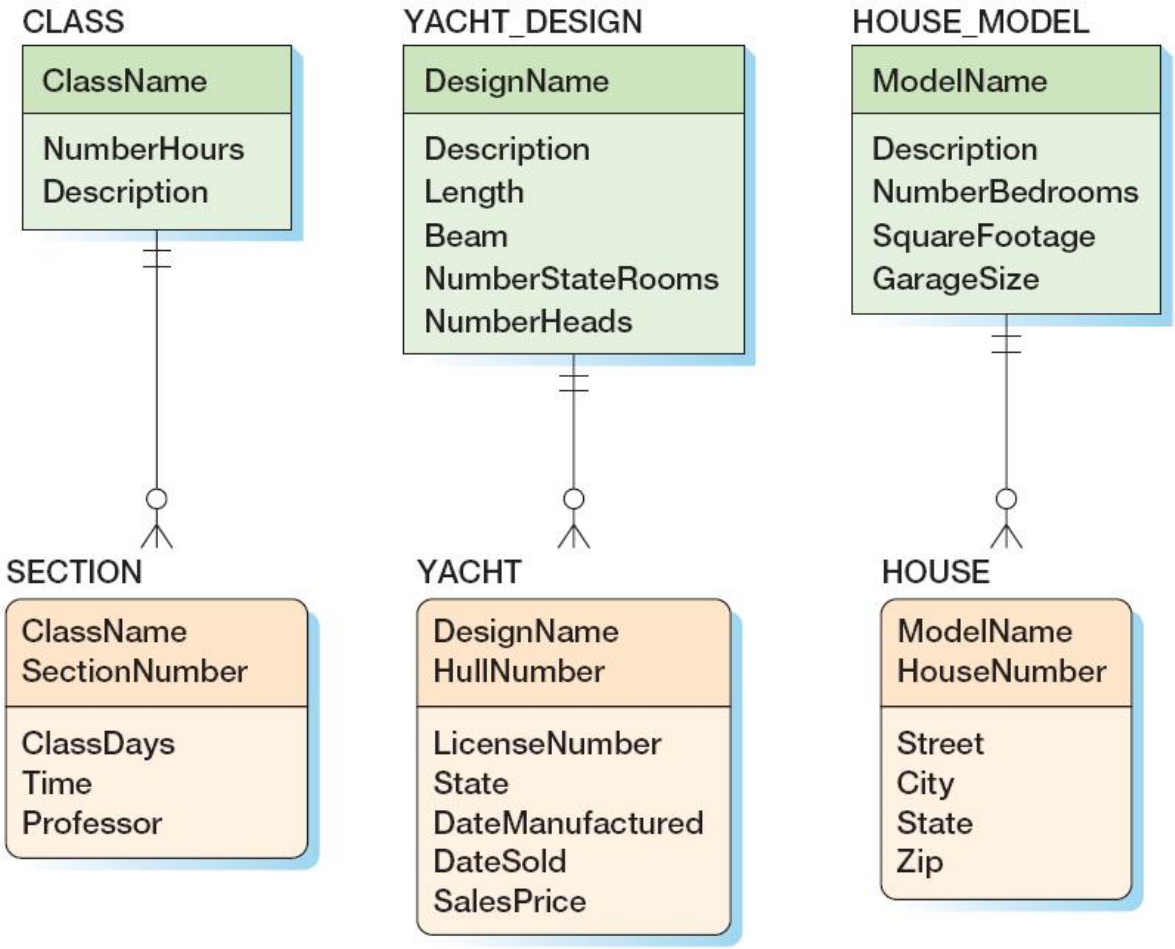
ID-Dependent Relationships: Composite Multivalued Attributes



ID-Dependent Relationships: The Archetype/Instance Pattern

- The **archetype/instance pattern** occurs when the ID-dependent child entity is the physical manifestation (instance) of an abstract or logical parent.
  - PAINTING : PRINT
  - CLASS : SECTION
  - YACHT_DESIGN: YACHT
  - HOUSE_MODEL: HOUSE

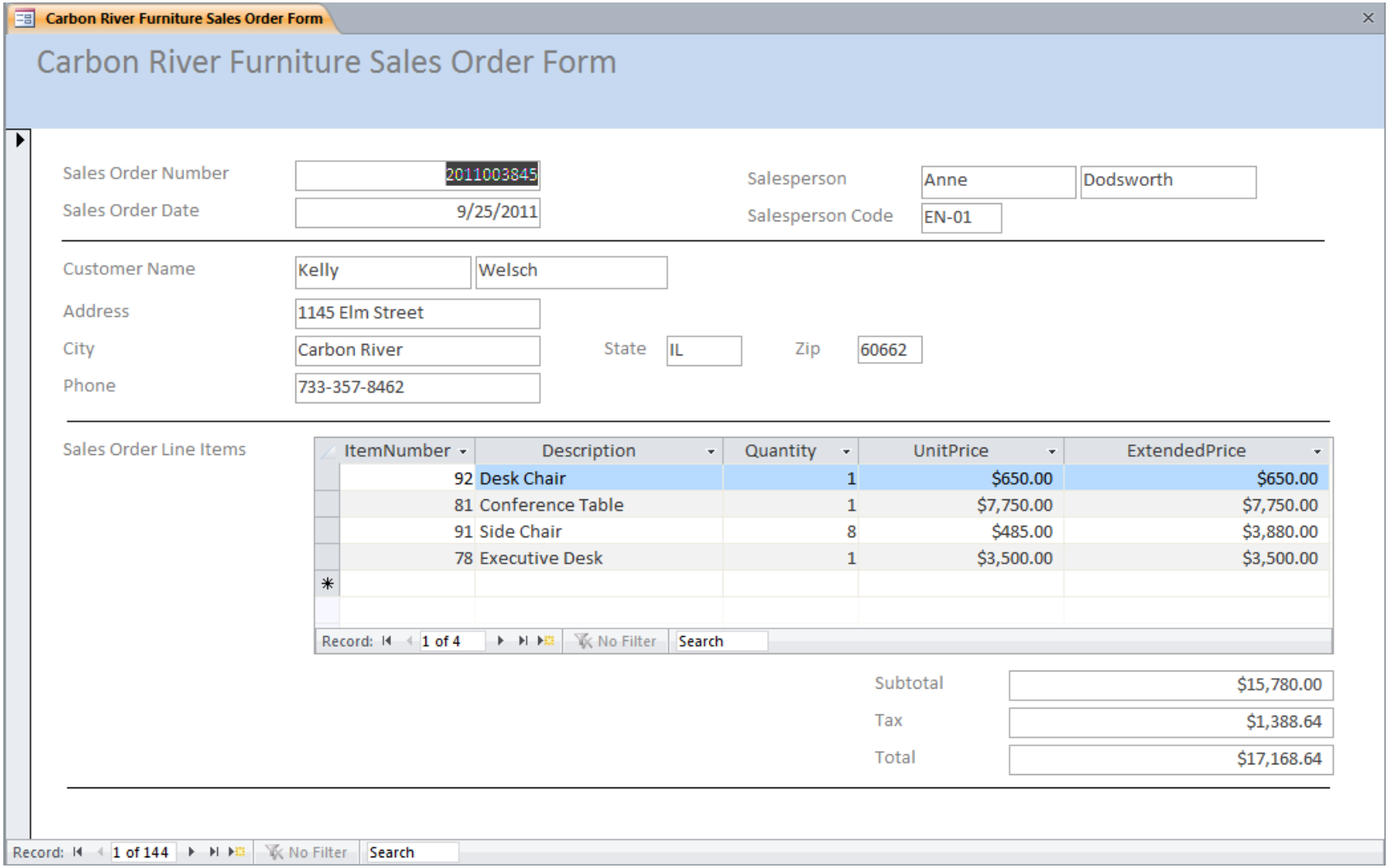
ID-Dependent Relationships: The Archetype/Instance Pattern



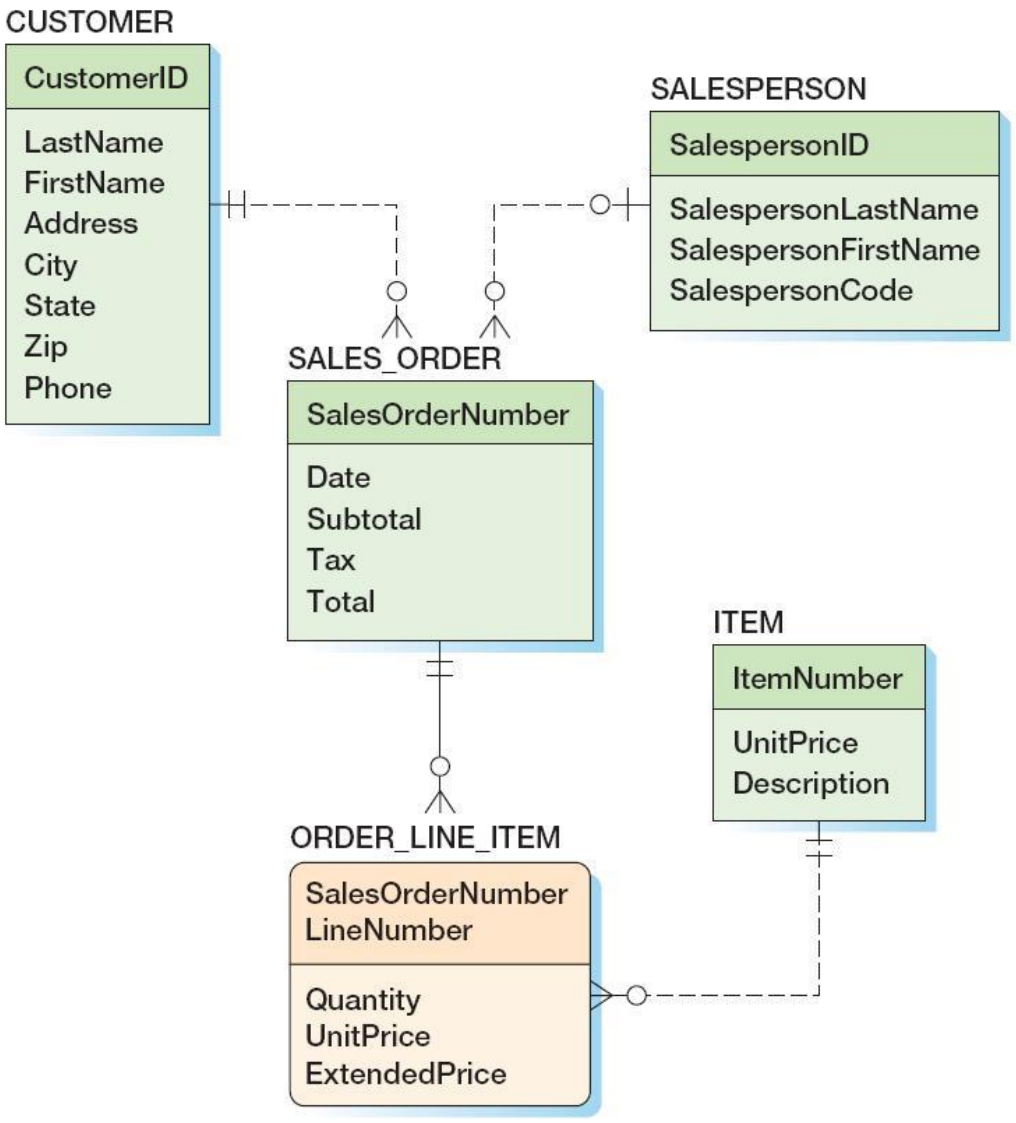
Mixed Patterns

- The Line-Item Pattern
- Other Patterns
- The For-Use-By Pattern
- Recursive Patterns

The Line-Item Pattern



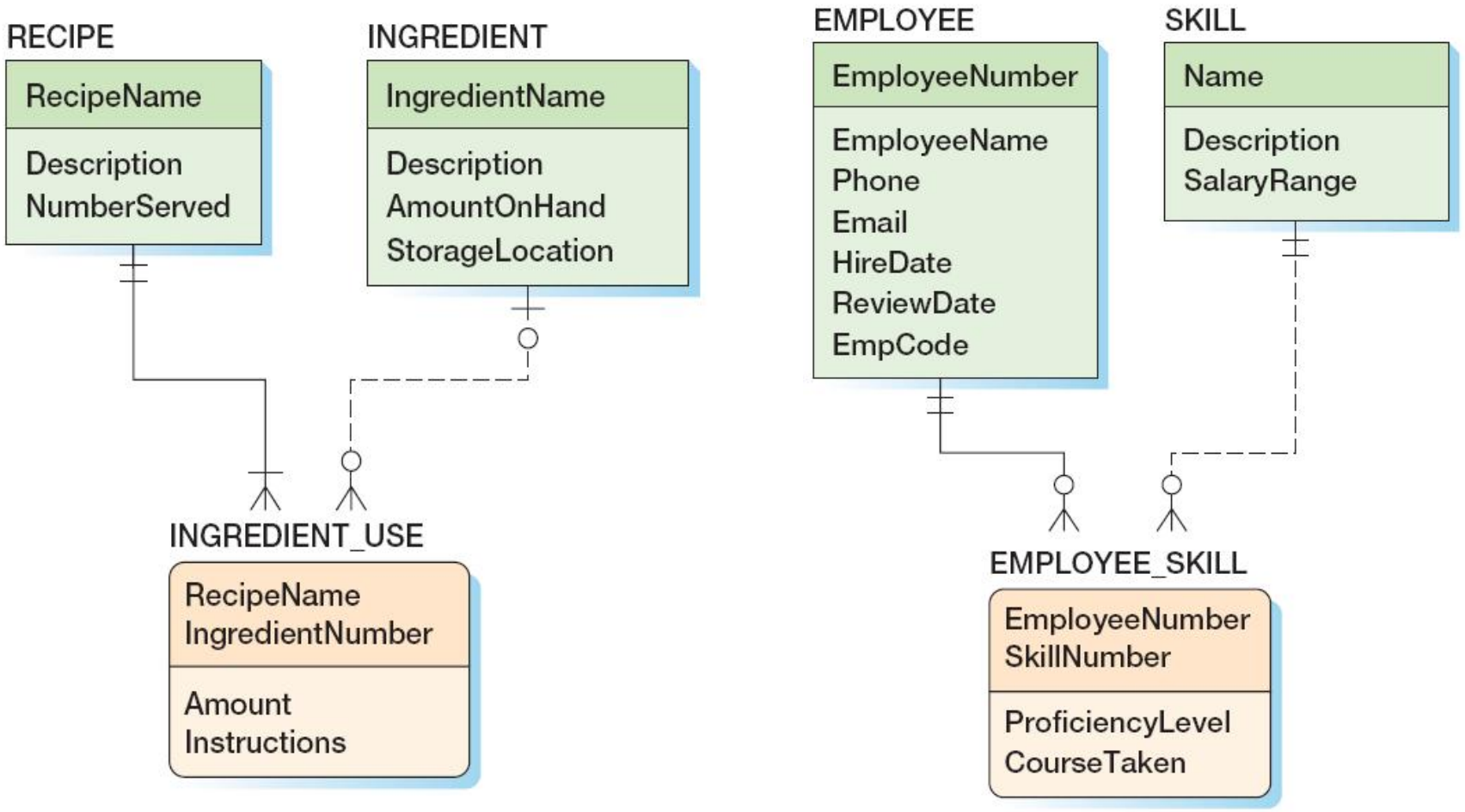
The Line-Item Pattern



Other Mixed Patterns

- Look for a mixed pattern where:
  - *A strong entity has a multivalued composite group*
  - One of the elements of the composite group is an identifier of another strong entity.

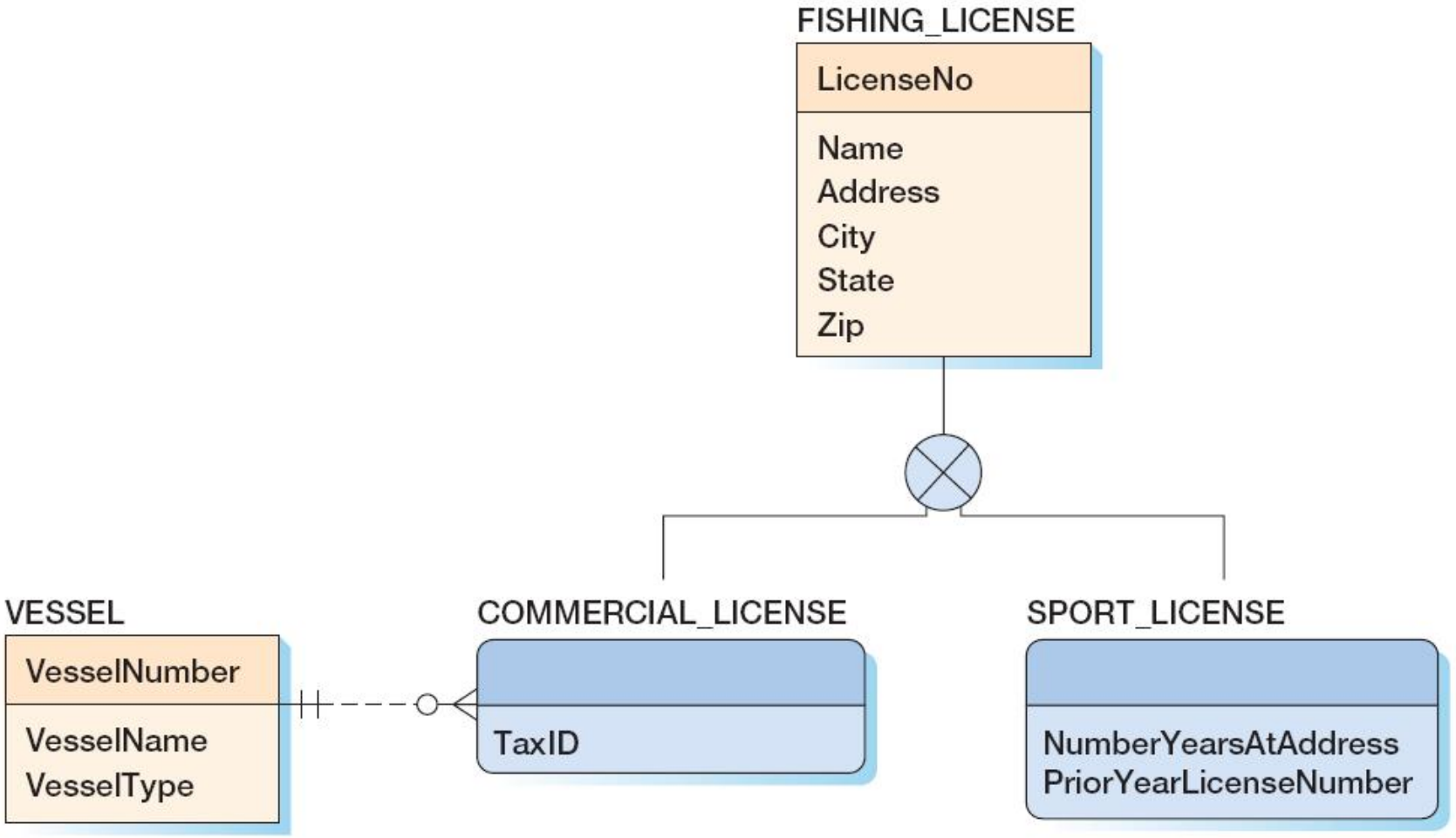
Other Mixed Patterns



The For-Use-By Pattern

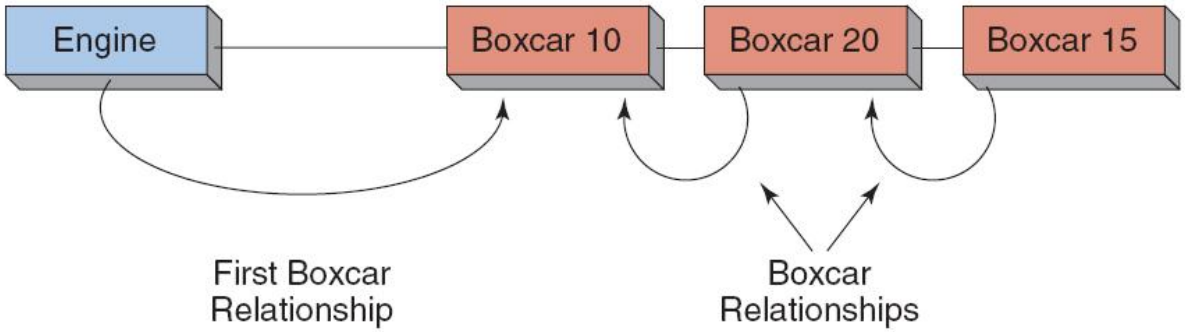
Resident Fishing License 2011 Season *State of Washington*					**License No:** **03-1123432**
Name:					
Street:					
City:		State:		Zip:	
For Use by Commercial Fishers Only			For Use by Sport Fishers Only		
Vessel Number:			Number Years at This Address:		
Vessel Name:			Prior Year License Number:		
Vessel Type:					
Tax ID:					

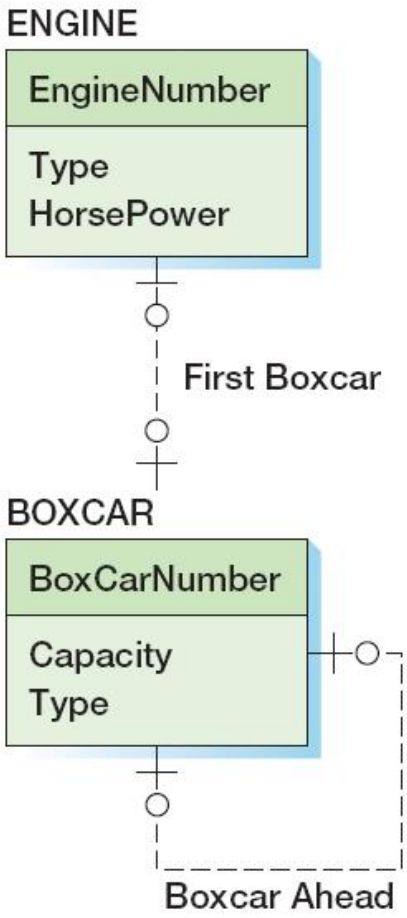
The For-Use-By Pattern



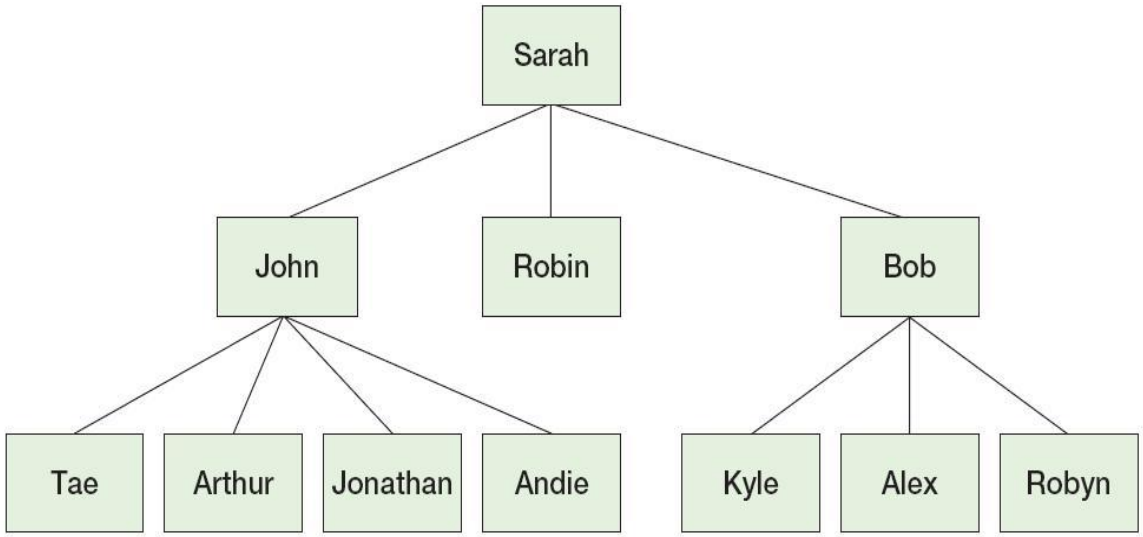
Recursive Relationships

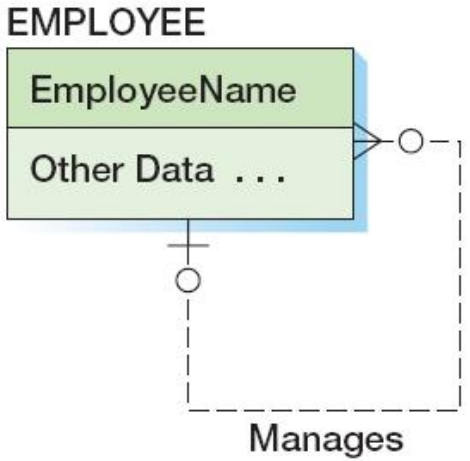
- A **recursive relationship** occurs when an entity has a relationship to itself





1:N Recursive Relationship





N:M Recursive Relationship