DATA REQUIREMENTS

ITACS 5203, Unit 6

Introduction
Data Requirements
E-R Model
  Conceptual
  E-R Syntax
  Types
  Business Rules
  Packaged Models
Class Model
  Domain Model
  Relationships
Security Requirements

LEARNING OBJECTIVES

Describe options for designing and conducting interviews and develop a plan for conducting an interview to determine system requirements.

Explain the advantages and pitfalls of observing workers and analyzing business documents to determine system requirements.

Explain how computing can provide support for requirements determination.

Participate in and help plan a Joint Application Design session.

Use prototyping during requirements determination.

Describe contemporary approaches to requirements determination.

Understand how requirements determination techniques apply to the development of electronic commerce applications.

Introduction

- Data Requirements
- E-R Model
  - Conceptual
  - E-R Syntax
  - Types
  - Business Rules
  - Packaged Models
- Class Model
  - Domain Model
  - Relationships
- Security Requirements

TOPICS

8. Structured System Data Requirements
8.1. Conceptual Data Modeling
8.1.1. E-R Diagraming
8.1.2. Super/Sub Types
8.1.3. Business Rules
8.1.4. Databases

3.1. System Development Methodologies SDLC Phases
3.1.1.1. Acquisition
3.1.1.1.1. Design
3.3.3.3.2.3. Entity Relationship Diagrams

1.9. SDLS and SSD
1.9.1. Incident Response Plan
1.9.2. Final Security Review
1.9.3. Incorporating OWASP and MS SDLC

Introduction

Data Requirements

E-R Model

- Conceptual
- E-R Syntax
- Types
- Business Rules
- Packaged Models

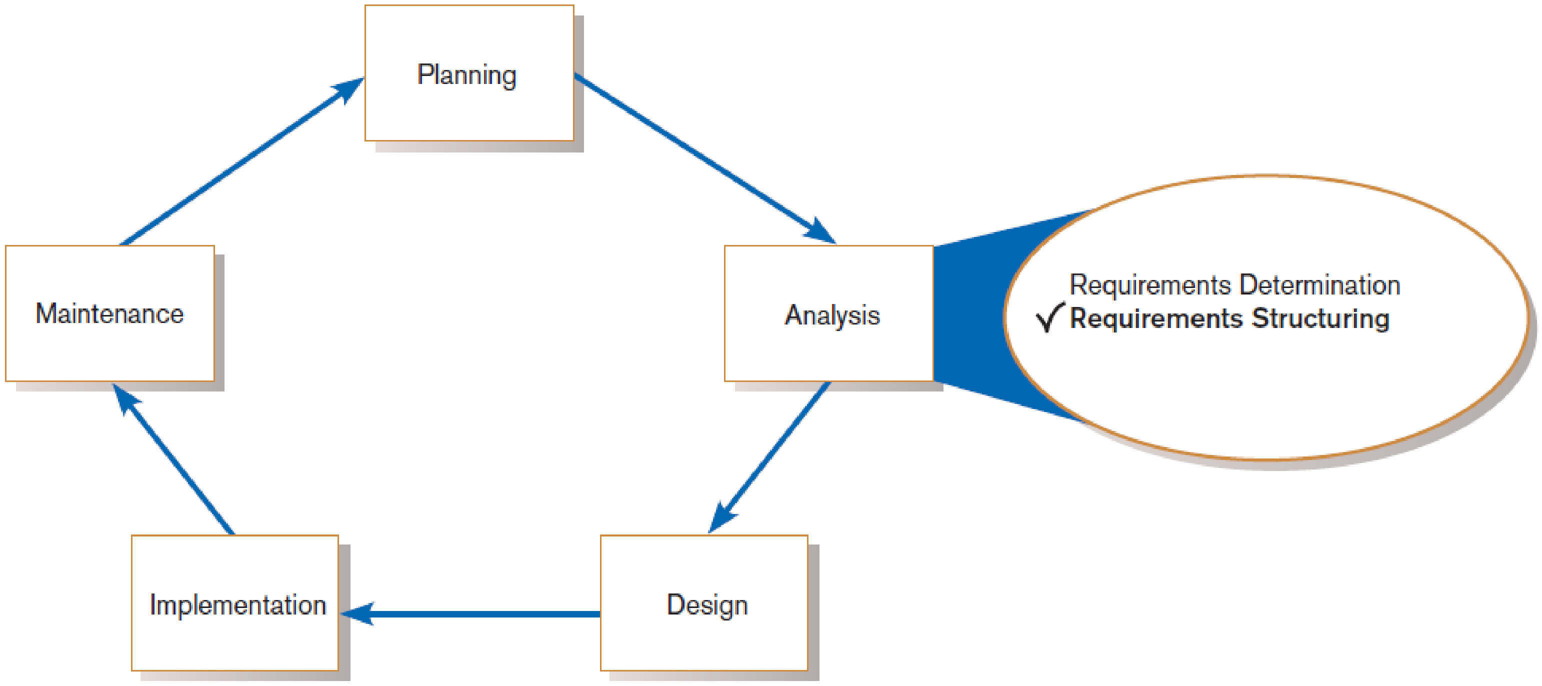
Class Model

- Domain Model
- Relationships

Security Requirements

TOPICS

Main goal: to discover the key data contained in the problem domain and to build a structural model of the objects



Introduction

Data Requirements

E-R Model

- Conceptual
- E-R Syntax
- Types
- Business Rules
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements

GATHERING INFORMATION FOR CONCEPTUAL DATA MODELS

Two perspectives on data modeling:

- *Top-down approach* for a data model is derived from an intimate understanding of the business.
- *Bottom-up approach* for a data model is derived by reviewing specifications and business documents.

THE PROBLEM DOMAIN

Problem domain—the specific area (or domain) of the users' business need that is within the scope of the new system.

"Things" are those items users work with when accomplishing tasks that need to be remembered

Examples of "Things" are products, sales, shippers, customers, invoices, payments, etc.

Two common techniques to identify "things":

- Brainstorming Technique
  - Use a checklist of all of the usual types of things typically found and brainstorm to identify domain classes of each type
- Noun Technique
  - Identify all of the nouns that come up when the system is described and determine if each is a domain class, an attribute, or not something we need to remember

Introduction

Data Requirements

E-R Model

- Conceptual
- E-R Syntax
- Types
- Business Rules
- Packaged Models

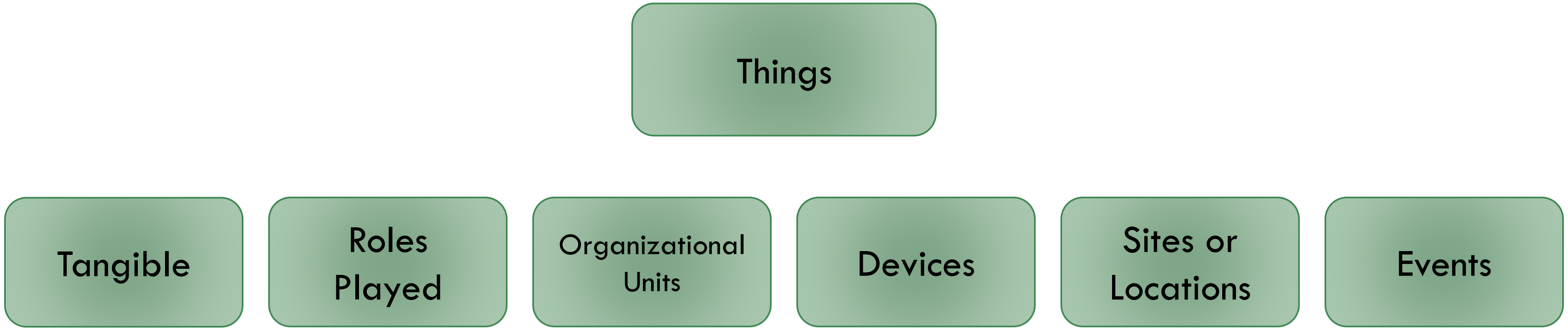
Class Model

- Domain Model
- Relationships

Security Requirements

THE PROBLEM DOMAIN

Are there any:



Things

Tangible	Roles Played	Organizational Units	Devices	Sites or Locations	Events
Airplane	Employee	Division	Sensor	Warehouse	Flight
Book	Customer	Department	Timer	Office	Call
Vehicle	Doctor	Section	Controller	Factory	Logon
Document	Patient	Task Force	Machine	Store	Logoff
Form	User	Workgroup	Sorter	Desktop	Contract
	Sys Admin		Printer		Purchase Order
			Container		Payment

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

BRAINSTORMING TECHNIQUE

1. Identify a user and a set of use cases
2. Brainstorm with the user to identify things involved when carrying out the use case—that is, things about which information should be captured by the system.
3. Use the types of things (categories) to systematically ask questions about potential things, such as the following:
    1. Are there any tangible things you store information about?
    2. Are there any locations involved?
    3. Are there roles played by people that you need to remember?
4. Continue to work with all types of users and stakeholders to expand the brainstorming list
5. Merge the results, eliminate any duplicates, and compile an initial list

Introduction

Data Requirements

E-R Model

- Conceptual
- E-R Syntax
- Types
- Business Rules
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements

NOUN TECHNIQUE

A technique to identify problem domain classes (things) by finding, classifying, and refining a list of nouns that come up in in discussions or documents

Popular technique. Systematic.

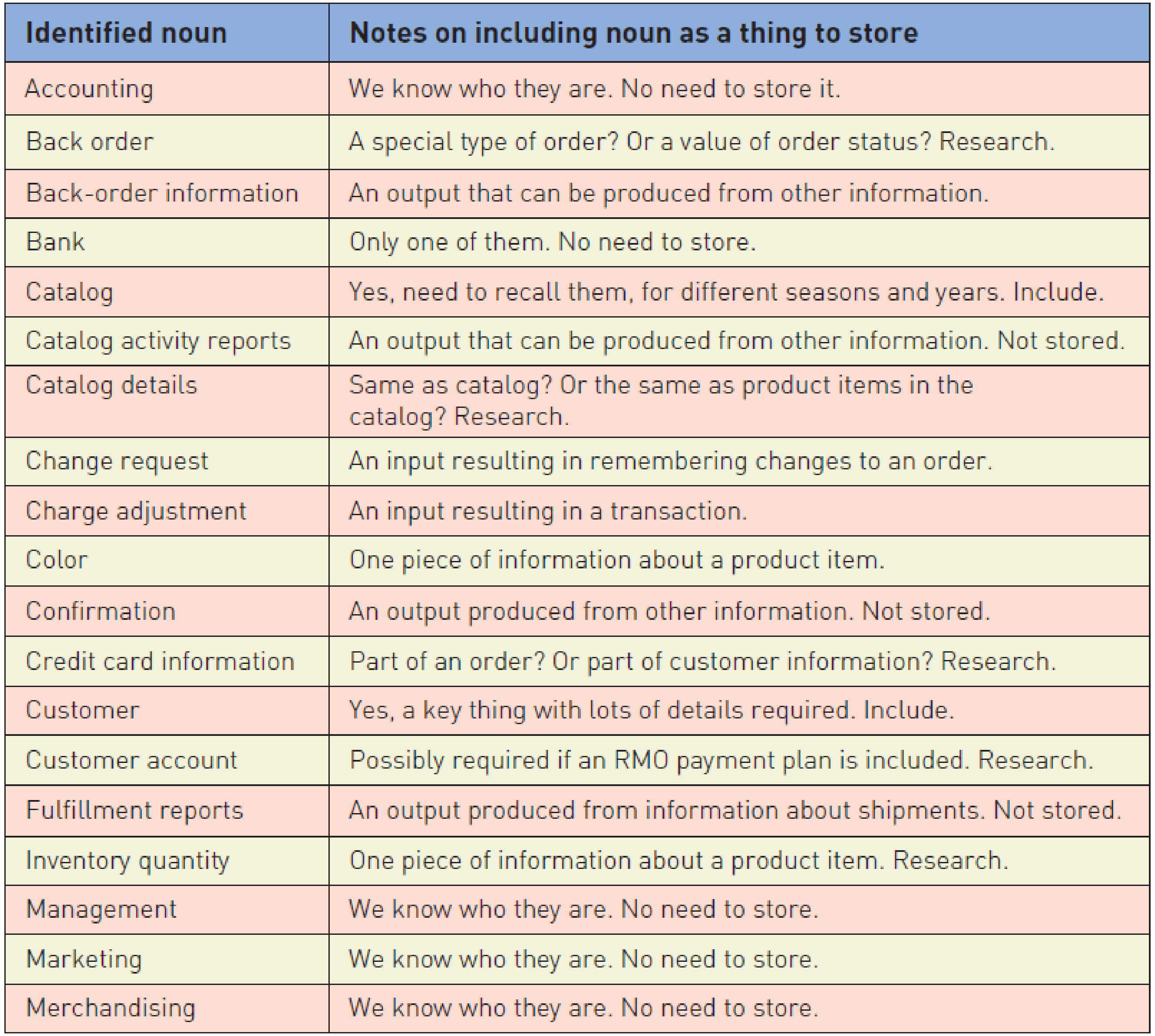
Does end up with long lists and many nouns that are not things that need to be stored by the system

Difficulty identifying synonyms and things that are really attributes

Good place to start when there are no users available to help brainstorm

- Introduction
- **Data Requirements**
- E-R Model
  - Conceptual
  - E-R Syntax
  - Types
  - Business Rules
  - Packaged Models
- Class Model
  - Domain Model
  - Relationships
- Security Requirements

NOUN TECHNIQUE EXAMPLE



Identified noun	Notes on including noun as a thing to store
Accounting	We know who they are. No need to store it.
Back order	A special type of order? Or a value of order status? Research.
Back-order information	An output that can be produced from other information.
Bank	Only one of them. No need to store.
Catalog	Yes, need to recall them, for different seasons and years. Include.
Catalog activity reports	An output that can be produced from other information. Not stored.
Catalog details	Same as catalog? Or the same as product items in the catalog? Research.
Change request	An input resulting in remembering changes to an order.
Charge adjustment	An input resulting in a transaction.
Color	One piece of information about a product item.
Confirmation	An output produced from other information. Not stored.
Credit card information	Part of an order? Or part of customer information? Research.
Customer	Yes, a key thing with lots of details required. Include.
Customer account	Possibly required if an RMO payment plan is included. Research.
Fulfillment reports	An output produced from information about shipments. Not stored.
Inventory quantity	One piece of information about a product item. Research.
Management	We know who they are. No need to store.
Marketing	We know who they are. No need to store.
Merchandising	We know who they are. No need to store.

NOUN TECHNIQUE STEPS

1. Using the use cases, actors, and other information about the system— including inputs and outputs—identify all nouns.
   - For the RMO CSMS, the nouns might include customer, product item, sale, confirmation, transaction, shipping, bank, change request, summary report, management, transaction report, accounting, back order, back order notification, return, return confirmation…
2. Using other information from existing systems, current procedures, and current reports or forms, add items or categories of information needed.
   - For the RMO CSMS, these might include price, size, color, style, season, inventory quantity, payment method, and shipping address.
3. As this list of nouns builds, refine it. Ask these questions about each noun to help you decide whether you should include it:
   - Is it a unique thing the system needs to know about?
   - Is it inside the scope of the system I am working on?
   - Does the system need to remember more than one of these items?

   Ask these questions to decide to exclude it:
   - Is it really a synonym for some other thing I have identified?
   - Is it really just an output of the system produced from other information I have identified?
   - Is it really just an input that results in recording some other information I have identified?

   Ask these questions to research it:
   - Is it likely to be a specific piece of information (attribute) about some other thing I have identified?
   - Is it something I might need if assumptions change?
4. Create a master list of all nouns identified and then note whether each one should be included, excluded, or researched further.
5. Review the list with users, stakeholders, and team members and then define the list of things in the problem domain.

- Introduction
- **Data Requirements**
- E-R Model
  - Conceptual
  - E-R Syntax
  - Types
  - Business Rules
  - Packaged Models
- Class Model
  - Domain Model
  - Relationships
- Security Requirements

MUST DETERMINE:

Requirements Determination Questions for Data Modeling:

- What are subjects/objects of the business?
  - Data entities and descriptions
- What unique characteristics distinguish between subjects/objects of the same type?
  - Primary keys
- What characteristics describe each subject/object?
  - Attributes and secondary keys
- How do you use the data?
  - Security controls and user access privileges
  - Who knows the meaning of the data?
- Over what period of time are you interested in the data?
  - Cardinality and time dimensions
- Are all instances of each object the same?
  - Supertypes, subtypes, and aggregations
- What events occur that imply associations between objects?
  - Relationships and cardinalities
- Are there special circumstances that affect the way events are handled?
  - Integrity rules, minimum and maximum cardinalities, time dimensions

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

DELIVERABLES AND OUTCOME

Entity-relationship (E-R) diagram or UML class diagram

- Entities (or classes) – categories of data, represented as rectangles
- Relationships (or associations) – lines between the entities

Set of entries about data objects to be stored in repository project dictionary, or data modeling software

- Repository links data, process, and logic models of an information system.
- Data elements included in the data flow diagram (DFD) must appear in the data model and vice versa.
- Each data store in a process model must relate to business objects represented in the data model.

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

CONCEPTUAL DATA MODELING PROCESS

Develop a data model for the current system.

Develop (or purchase) a new conceptual data model that includes all requirements of the new system.

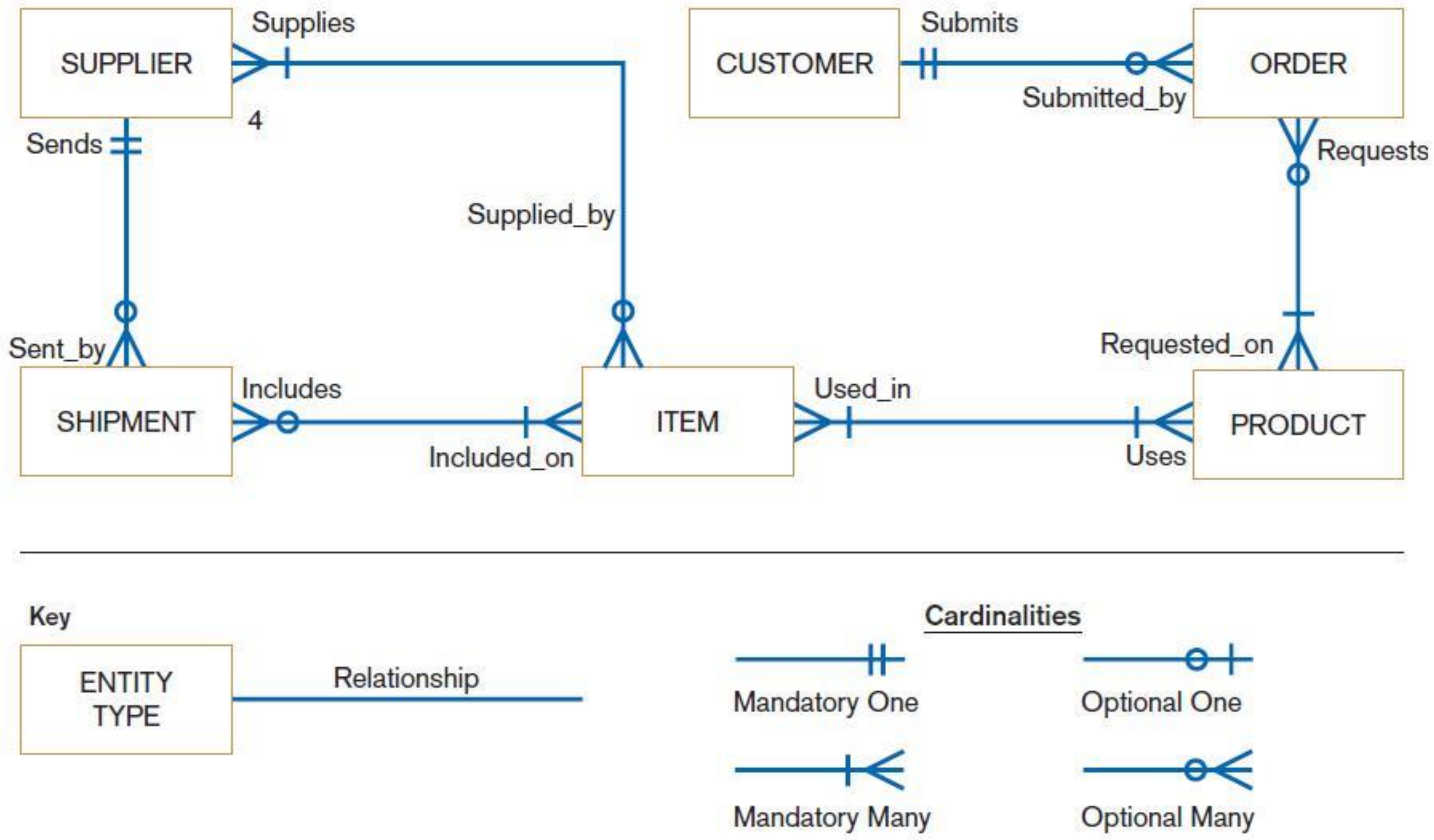
In the design stage, the conceptual data model is translated into a physical design.

Project repository links all design and data modeling steps performed during SDLC.

The E-R model is expressed in terms of:

- Data entities in the business environment.
- Relationships or associations among those entities.
- Attributes or properties of both the entities and their relationships.

ENTITY RELATIONSHIP MODELING



DELIVERABLES AND OUTCOME

Entity-Relationship data model (**E-R model**): a detailed, logical representation of the entities, associations and data elements for an organization or business area

Entity-relationship diagram (**E-R diagram**): a graphical representation of an E-R model

E-R VERNACULAR

Entity: a person, place, object, event or concept in the user environment about which data is to be maintained

Entity type: collection of entities that share common properties or characteristics

Entity instance: single occurrence of an entity type

Introduction

Data Requirements

E-R Model

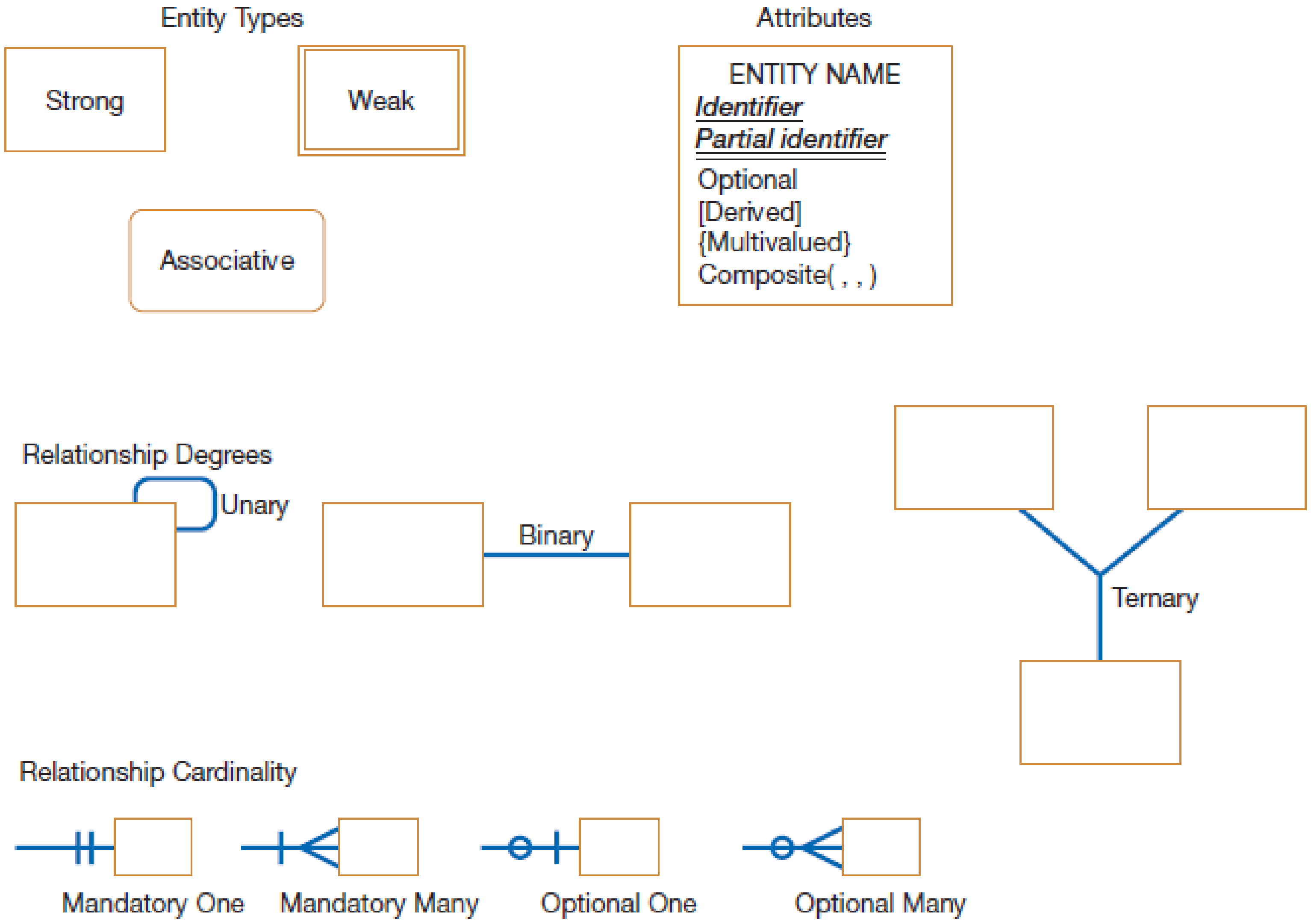
- Conceptual
- **E-R Syntax**
- Types
- Business Rules
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements

E-R VERNACULAR



Entity Types

Strong

Weak

Associative

Attributes

ENTITY NAME

Identifier

Partial identifier

Optional

[Derived]

{Multivalued}

Composite(, ,)

Relationship Degrees

Unary

Binary

Ternary

Relationship Cardinality

Mandatory One

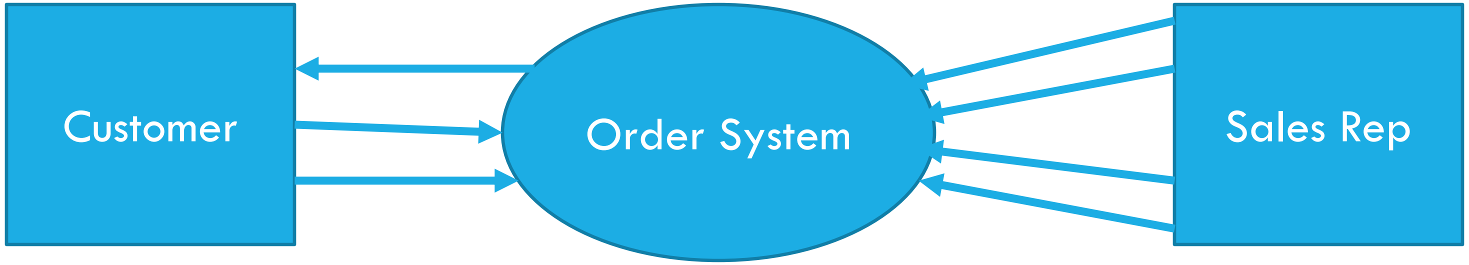
Mandatory Many

Optional One

Optional Many

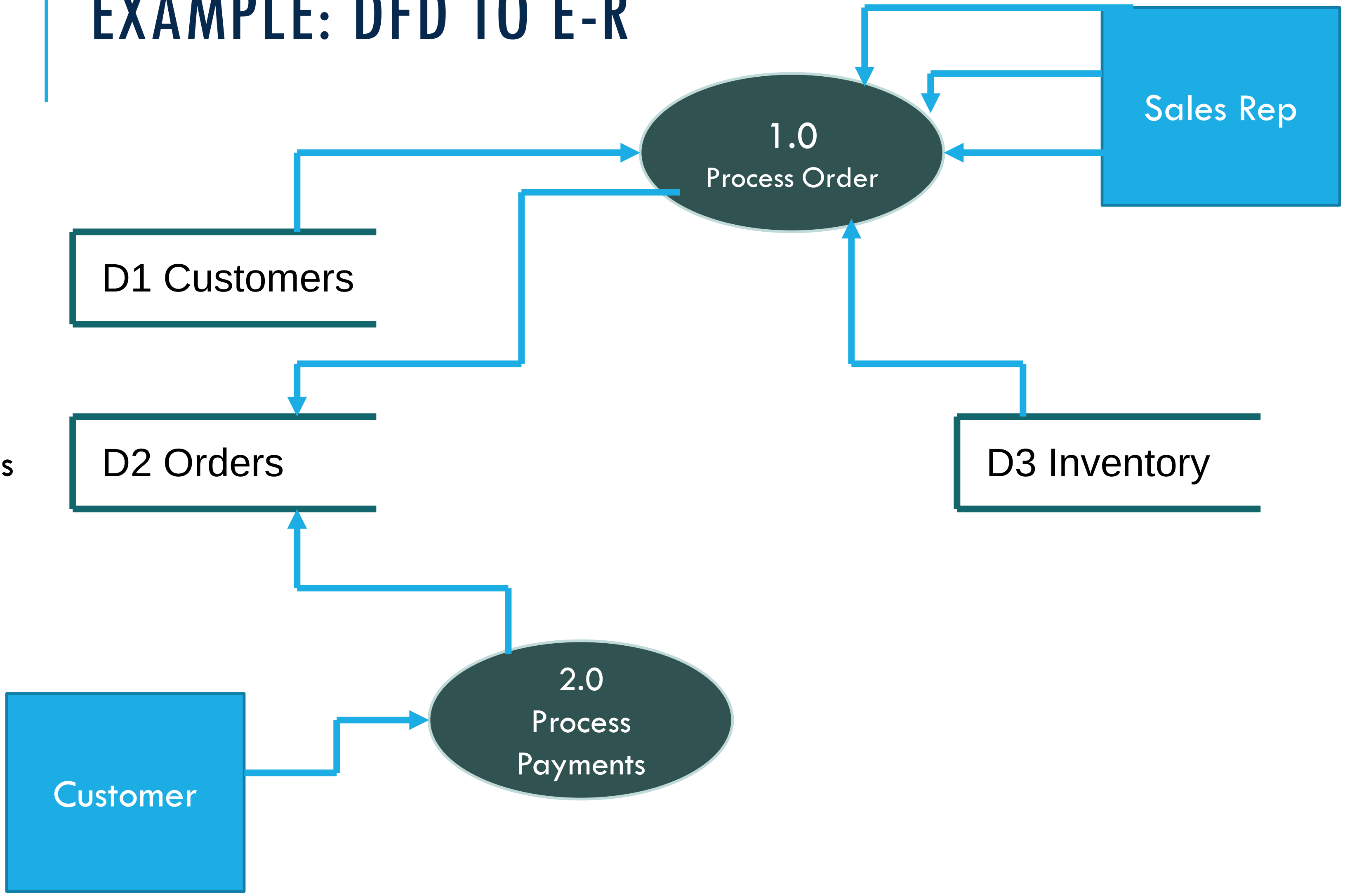
Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

EXAMPLE: DFD TO E-R



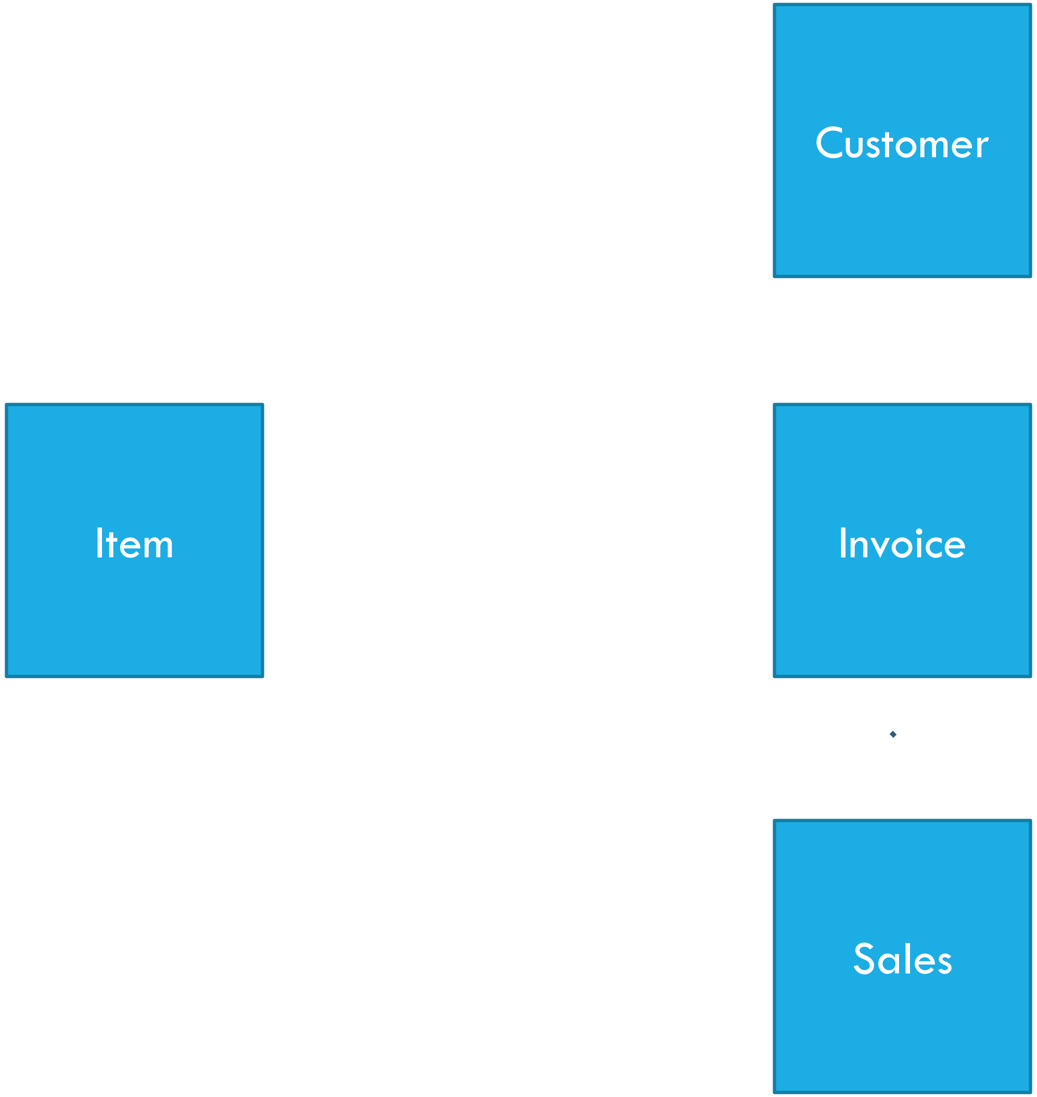
Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

EXAMPLE: DFD TO E-R



Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

EXAMPLE: DFD TO E-R



Customer

Item

Invoice

Sales

Introduction

Data Requirements

E-R Model

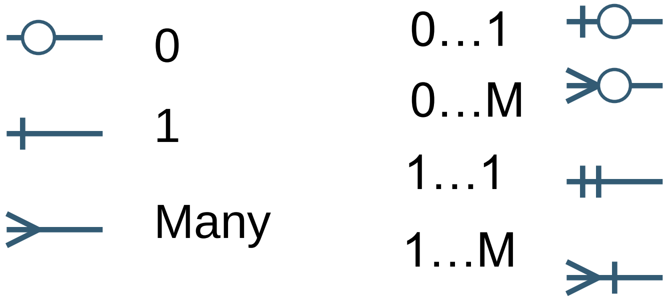
- Conceptual
- **E-R Syntax**
- Types
- Business Rules
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements

CARDINALITY



0

1

Many

0…1

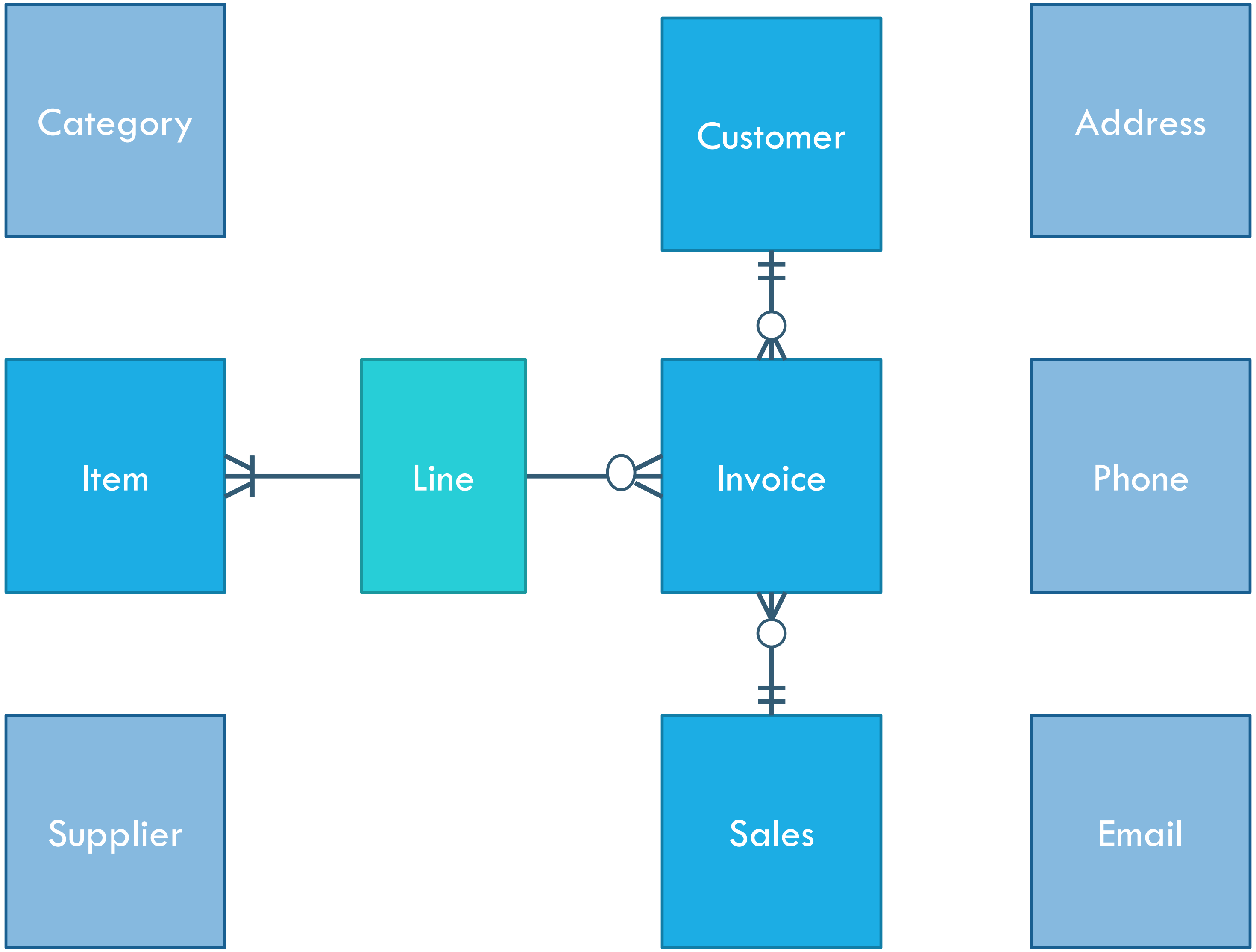
0…M

1…1

1…M

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

EXAMPLE: DFD TO E-R



Introduction

Data Requirements

E-R Model

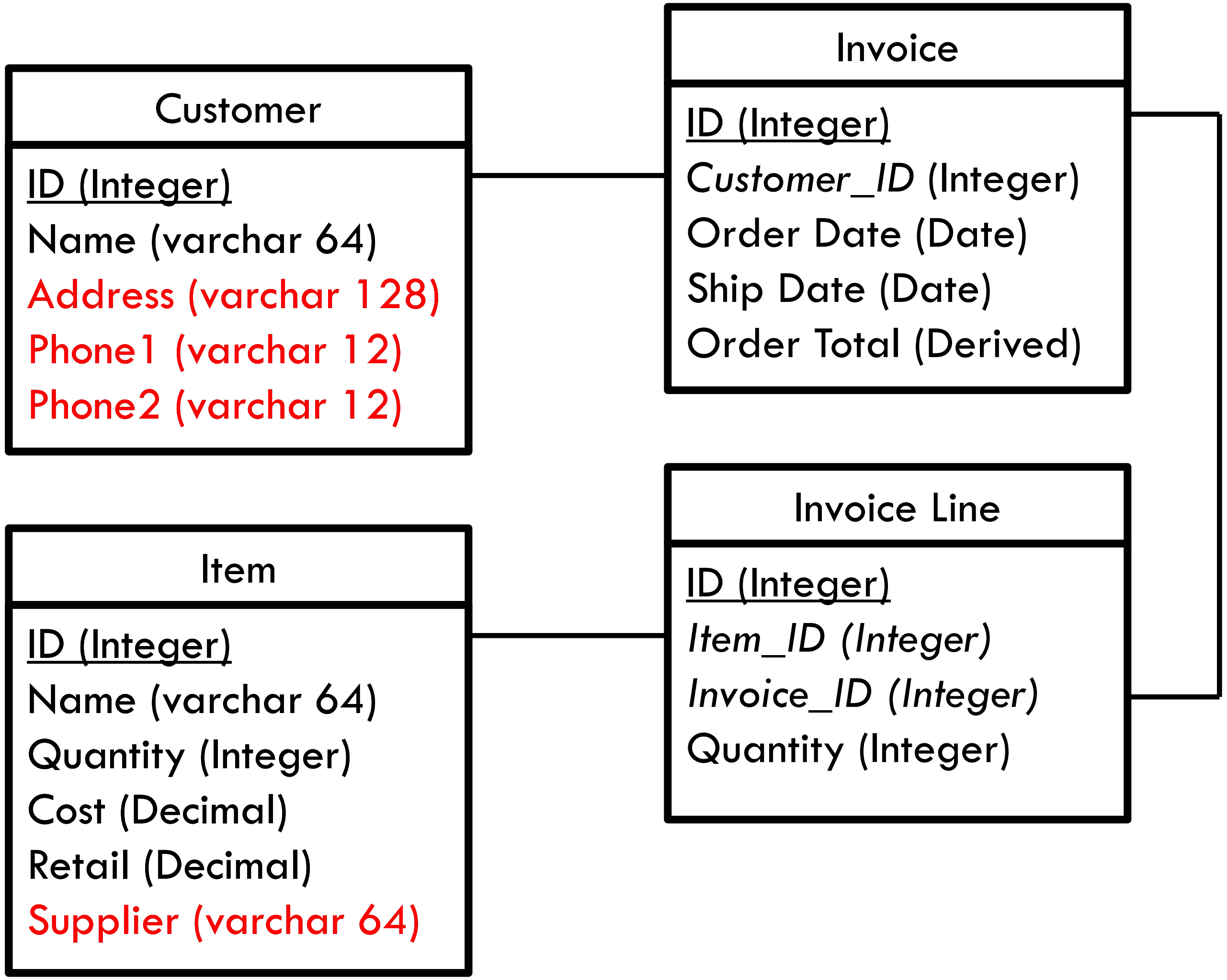
- Conceptual
- **E-R Syntax**
- Types
- Business Rules
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements

EXAMPLE: DFD TO E-R TO PHYSICAL DESIGN?



Customer

ID (Integer)
Name (varchar 64)
Address (varchar 128)
Phone1 (varchar 12)
Phone2 (varchar 12)

Invoice

ID (Integer)
Customer_ID (Integer)
Order Date (Date)
Ship Date (Date)
Order Total (Derived)

Item

ID (Integer)
Name (varchar 64)
Quantity (Integer)
Cost (Decimal)
Retail (Decimal)
Supplier (varchar 64)

Invoice Line

ID (Integer)
Item_ID (Integer)
Invoice_ID (Integer)
Quantity (Integer)

Introduction
Data Requirements
E-R Model
- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

Class Model
- Domain Model
- Relationships

Security Requirements

NAMING AND DEFINING ENTITY TYPES

An entity type name should be:

- *A singular noun.*
- *Descriptive and specific to the organization.*
- *Concise.*

Event entity type should be named for the *result of the event*, not the activity or process of the event.

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

NAMING AND DEFINING ENTITY TYPES

An entity type definition:

- Includes a statement of *what the unique characteristic(s) is (are) for each instance.*
- Makes clear *what entity instances are included and not included* in the entity type.
- Often includes a description of *when an instance of the entity type is created or deleted.*

NAMING AND DEFINING ENTITY TYPES

For some entity types the definition must specify:

- *When an instance might change into an instance of another entity type.*
- *What history is to be kept about entity instances.*

ATTRIBUTES

Attribute: a named property or characteristic of an entity that is of interest to the organization

- Naming an attribute: i.e. Vehicle_ID
- Place its name inside the rectangle for the associated entity in the E-R diagram.

- Introduction
- Data Requirements
- **E-R Model**
  - Conceptual
  - E-R Syntax
  - **Types**
  - Business Rules
  - Packaged Models
- Class Model
  - Domain Model
  - Relationships
- Security Requirements

NAMING AND DEFINING ATTRIBUTES

An attribute name is a *noun* and should be *unique.*

To make an attribute name unique and for clarity, *each attribute name should follow a standard format.*

Similar attributes of different entity types should use similar but distinguishing names.

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

NAMING AND DEFINING ATTRIBUTES

An attribute definition:

- *States what the attribute is and possibly why it is important.*
- Should make it clear *what is included and what is not included.*
- Contains any *aliases* or alternative names.
- States *the source of values for the attribute.*

Introduction
Data Requirements
E-R Model
  Conceptual
  E-R Syntax
  Types
  Business Rules
  Packaged Models
Class Model
  Domain Model
  Relationships
Security Requirements

NAMING AND DEFINING ATTRIBUTES

An attribute definition should indicate:

- *If a value for the attribute is required or optional.*
- *If a value for the attribute may change.*
- Any *relationships that attribute has with other attributes.*

Introduction
Data Requirements
E-R Model
- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

Class Model
- Domain Model
- Relationships

Security Requirements

CANDIDATE KEYS AND IDENTIFIERS

Candidate key: an attribute (or combination of attributes) that uniquely identifies each instance of an entity type

Identifier: a candidate key that has been selected as the unique, identifying characteristic for an entity type

CANDIDATE KEYS AND IDENTIFIERS

Selection rules for an identifier

- Choose a candidate key that will not change its value.
- Choose a candidate key that will never be null.
- Avoid using intelligent keys.
- Consider substituting single value surrogate keys for large composite keys.

OTHER ATTRIBUTE TYPES

Multivalued attribute: an attribute that may take on more than one value for each entity instance

Repeating group: a set of two or more multivalued attributes that are logically related

Introduction

Data Requirements

E-R Model

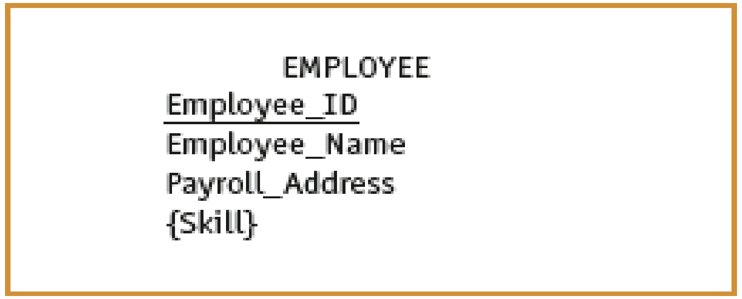
- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements

OTHER ATTRIBUTE TYPES



EMPLOYEE

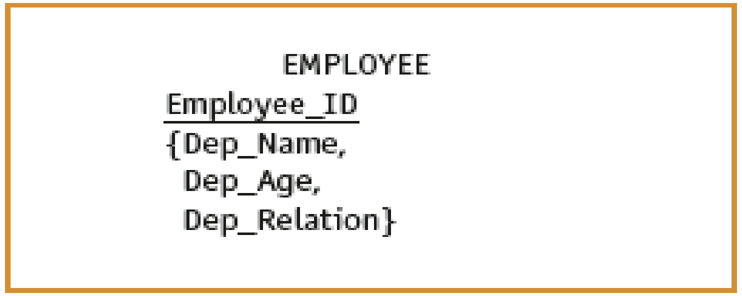
Employee_ID

Employee_Name

Payroll_Address

{Skill}

(a) Multivalued attribute skill

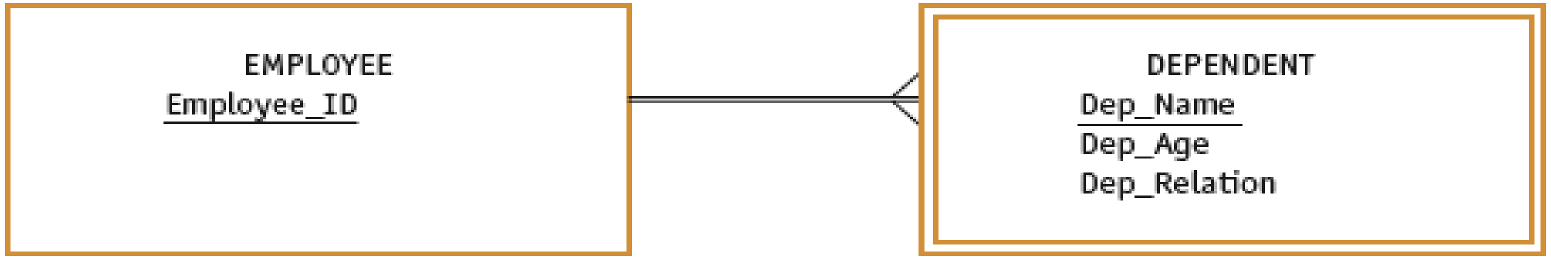


EMPLOYEE

Employee_ID

{Dep_Name,
 Dep_Age,
 Dep_Relation}

(b) Repeating group of dependent data



EMPLOYEE

Employee_ID

DEPENDENT

Dep_Name

Dep_Age

Dep_Relation

(c) Weak entity for dependent data

Multivalued attributes
and repeating groups

Introduction
Data Requirements
E-R Model
- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

Class Model
- Domain Model
- Relationships

Security Requirements

OTHER ATTRIBUTE TYPES

Required attribute: an attribute that must have a value for every entity instance

Optional attribute: an attribute that may not have a value for every entity instance

Composite attribute: an attribute that has meaningful component parts

Derived attribute: an attribute whose value can be computed from related attribute values

Introduction

Data Requirements

E-R Model

- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements

RELATIONSHIPS

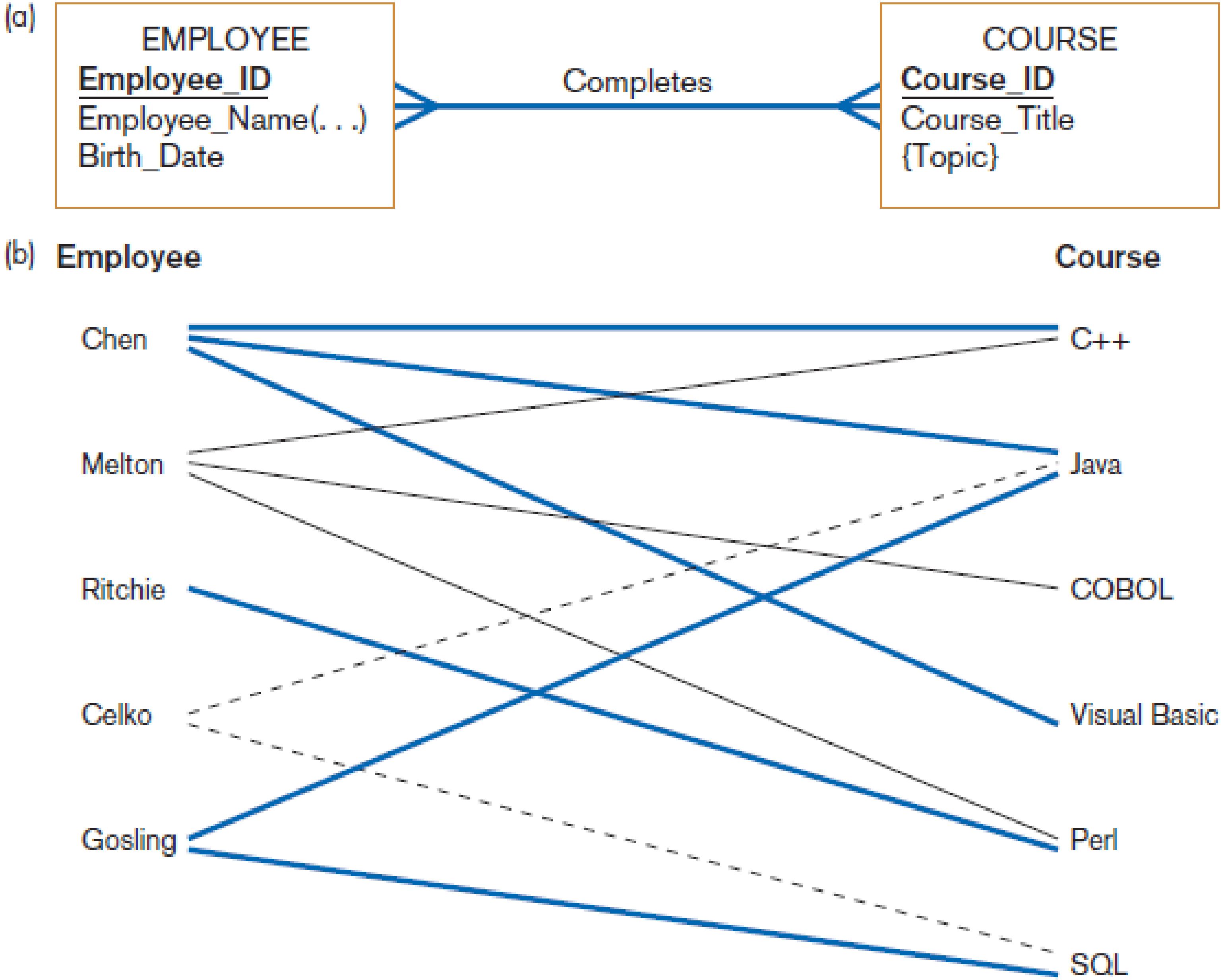
Relationship: an association between the instances of one or more entity types that is of interest to the organization

Degree: the number of entity types that participate in a relationship

Introduction
Data Requirements
E-R Model
- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

Class Model
- Domain Model
- Relationships

Security Requiremen



Relationship type and instances
(a) Relationship type (Completes)
(b) Relationship instances

RELATIONSHIPS

Unary relationship: a relationship between the instances of one entity type

- Also called a *recursive relationship*

Binary relationship: a relationship between instances of two entity types

- Most common type of relationship encountered in data modeling

Ternary relationship: a simultaneous relationship among instances of three entity types

RELATIONSHIPS

Cardinality: the number of instances of entity B that can (or must) be associated with each instance of entity A

Minimum Cardinality

- The minimum number of instances of entity B that may be associated with each instance of entity A

Maximum Cardinality

- The maximum number of instances of entity B that may be associated with each instance of entity A

Introduction

Data Requirements

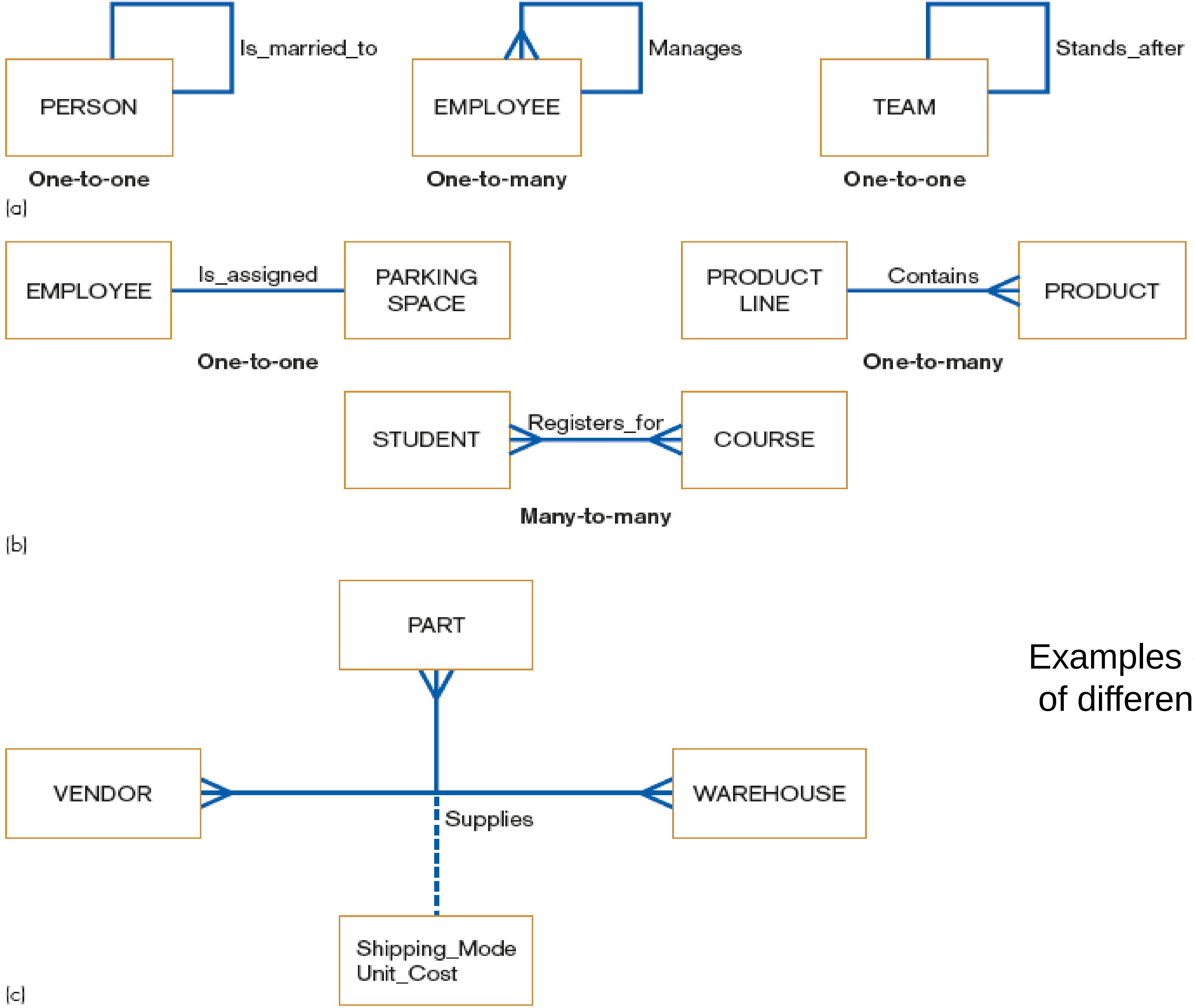
E-R Model

- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements



Examples of relationships of different degrees

Introduction

Data Requirements

E-R Model

- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements

RELATIONSHIPS

Mandatory vs. Optional Cardinalities

- Specifies whether an instance must exist or can be absent in the relationship

Introduction

Data Requirements

E-R Model

- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

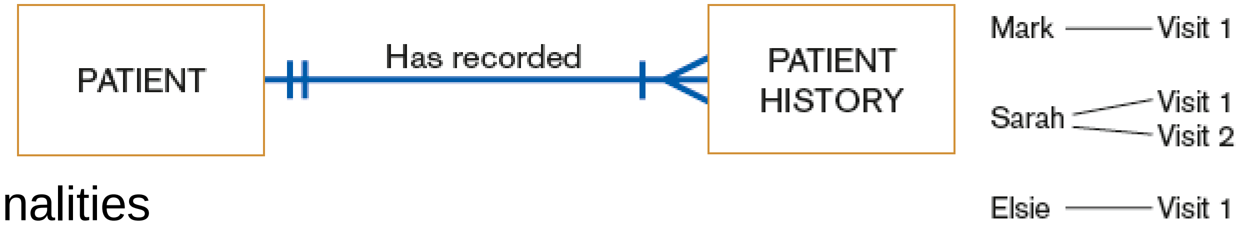
Class Model

- Domain Model
- Relationships

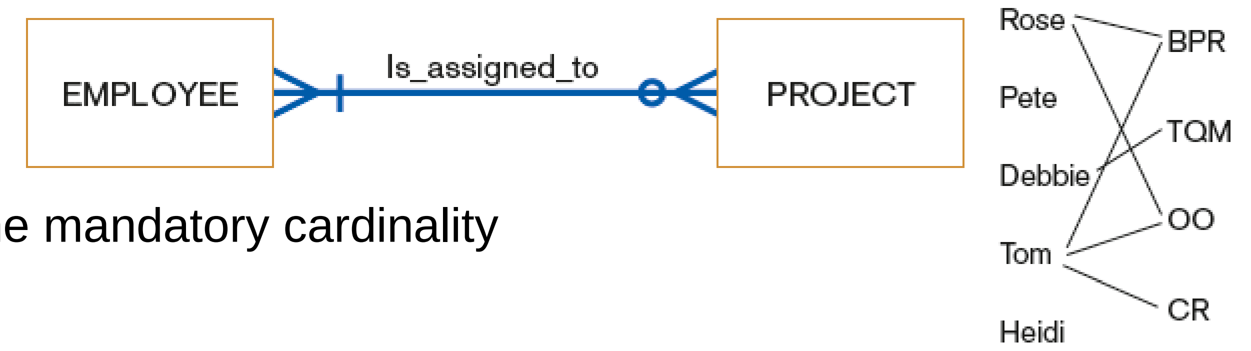
Security Requirements

RELATIONSHIPS

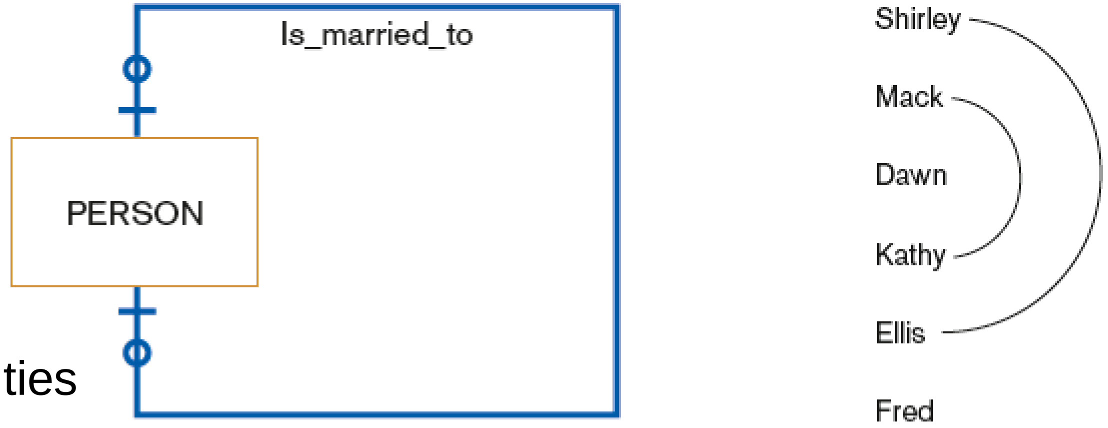
Examples of cardinality constraints



(a) Mandatory cardinalities



(b) One optional, one mandatory cardinality



(c) Optional cardinalities

Introduction
Data Requirements
E-R Model
- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

Class Model
- Domain Model
- Relationships

Security Requirements

NAMING AND DEFINING RELATIONSHIPS

A relationship name is a verb phrase; avoid vague names.

A relationship definition:

- Explains what action is to be taken and possibly why it is important.
- Gives examples to clarify the action.

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

NAMING AND DEFINING RELATIONSHIPS

A relationship definition should:

- Explain any optional participation.
- Explain the reason for any explicit maximum cardinality other than many.
- Explain any restrictions on participation in the relationship.
- Explain the extent of history that is kept in the relationship.
- Explain whether an entity instance involved in a relationship instance can transfer participation to another relationship instance.

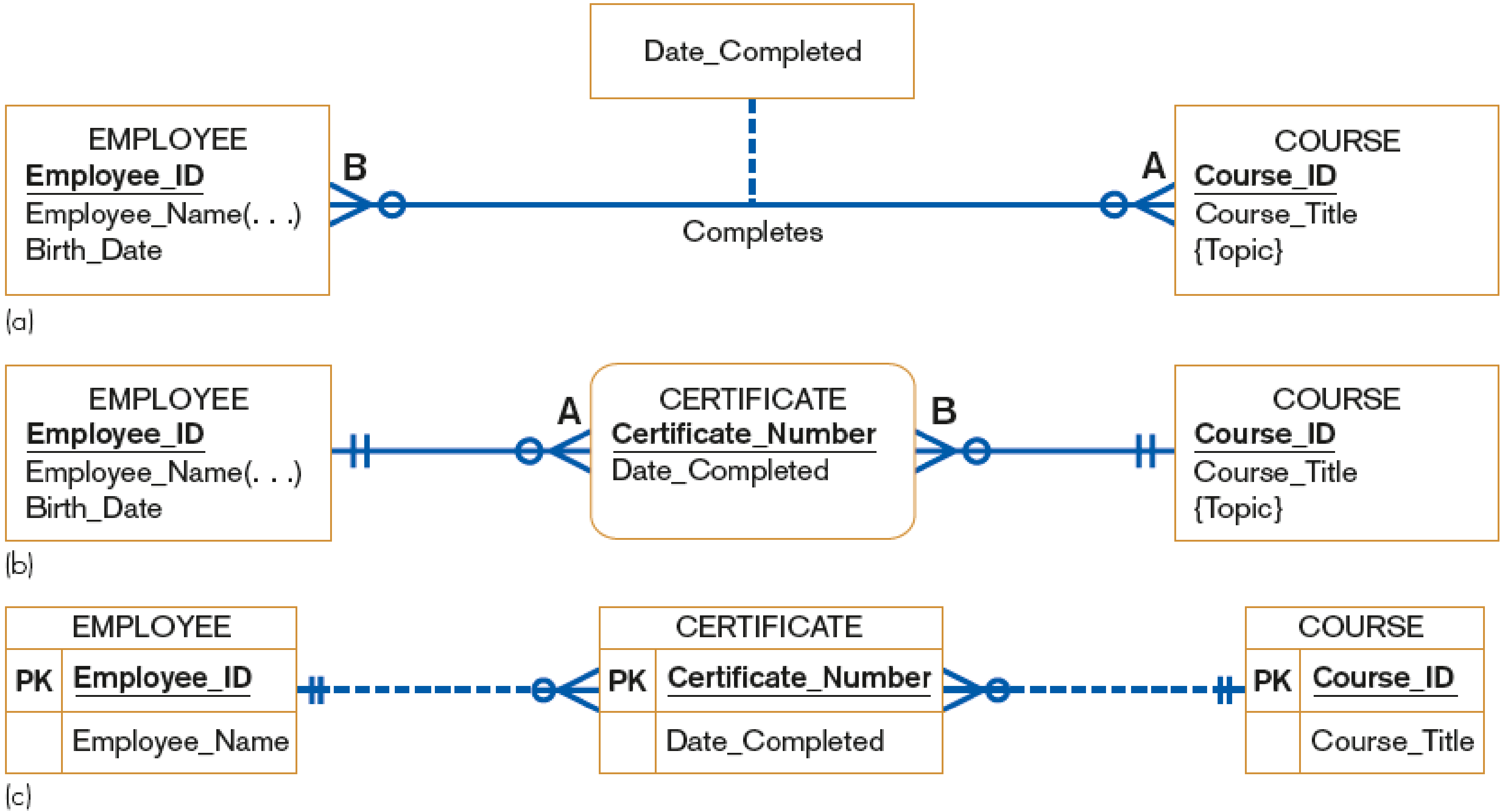
ASSOCIATIVE ENTITIES

Associative Entity: an entity type that associates the instances of one or more entity types and contains attributes that are peculiar to the relationship between those entity instances

The data modeler chooses to model the relationship as an entity type.

ASSOCIATIVE ENTITIES

An associative entity



An associative entity using Microsoft Visio®

REPRESENTING SUPERTYPES AND SUBTYPES

Subtype: a subgrouping of the entities in an entity type

- Is meaningful to the organization
- Shares common attributes or relationships distinct from other subgroupings

Supertype: a generic entity type that has a relationship with one or more subtypes

Introduction

Data Requirements

E-R Model

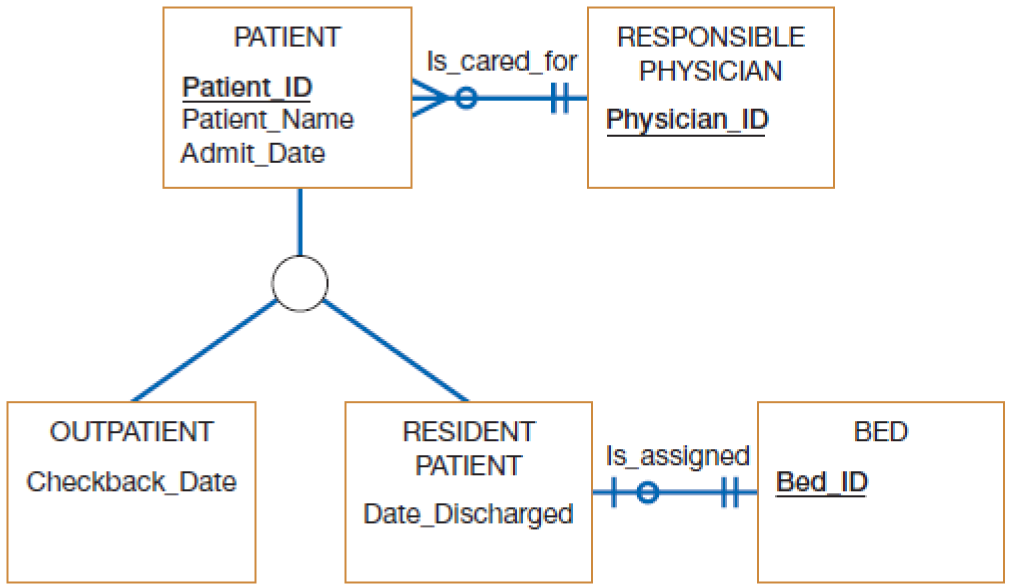
- Conceptual
- E-R Syntax
- **Types**
- Business Rules
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements

REPRESENTING SUPERTYPES AND SUBTYPES



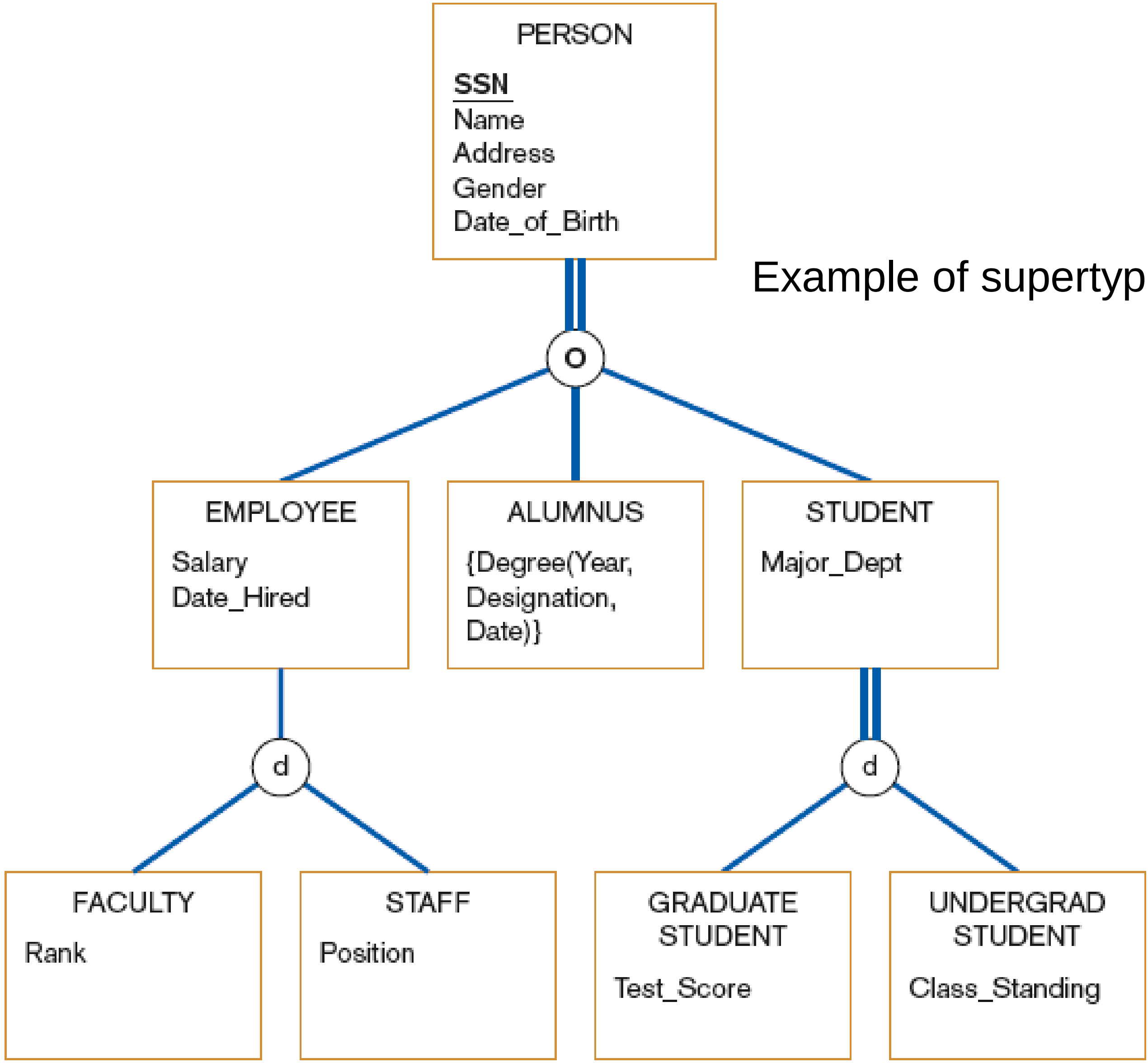
REPRESENTING SUPERTYPES AND SUBTYPES

Business Rules for Supertype/subtype Relationships:

- **Total specialization** specifies that each entity instance of the supertype must be a member of some subtype in the relationship.
- **Partial specialization** specifies that an entity instance of the supertype does not have to belong to any subtype, and may or may not be an instance of one of the subtypes.
- **Disjoint rule** specifies that if an entity instance of the supertype is a member of one subtype, it cannot simultaneously be a member of any other subtype.
- **Overlap rule** specifies that an entity instance can simultaneously be a member of two (or more) subtypes.

Introduction
Data Requirements
E-R Model
  Conceptual
  E-R Syntax
  Types
  Business Rules
  Packaged Models
Class Model
  Domain Model
  Relationships
Security Requirements

REP



Example of supertype/subtype hierarchy

Introduction
Data Requirements
E-R Model
- Conceptual
- E-R Syntax
- Types
- **Business Rules**
- Packaged Models

Class Model
- Domain Model
- Relationships

Security Requirements

BUSINESS RULES

Business rules: specifications that preserve the integrity of the logical data model

- Captured during requirements determination
- Stored in CASE repository as they are documented

BUSINESS RULES

Four basic types of business rules are:

- *Entity integrity*: unique, non-null identifiers
- *Referential integrity constraints*: rules governing relationships between entity types
- *Domains*: constraints on valid values for attributes
- *Triggering operations*: other business rules that protect the validity of attribute values

Introduction
Data Requirements
E-R Model
  Conceptual
  E-R Syntax
  Types
  Business Rules
  Packaged Models
Class Model
  Domain Model
  Relationships
Security Requirements

DOMAINS

Domain: the set of all data types and values that an attribute can assume

Several advantages

- Verify that the values for an attribute are valid
- Ensure that various data manipulation operations are logical
- Help conserve effort in describing attribute characteristics

Introduction

Data Requirements

E-R Model

- Conceptual
- E-R Syntax
- Types
- **Business Rules**
- Packaged Models

Class Model

- Domain Model
- Relationships

Security Requirements

TRIGGERING OPERATIONS

Trigger: an assertion or rule that governs the validity of data manipulation operations such as insert, update and delete

TRIGGERING OPERATIONS

Includes the following components:

- *User rule*: statement of the business rule to be enforced by the trigger
- *Event*: data manipulation operation that initiates the operation
- *Entity Name*: name of entity being accessed or modified
- *Condition*: condition that causes the operation to be triggered
- *Action*: action taken when the operation is triggered

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

PACKAGED DATA MODELS

Packaged data models provide generic models that can be customized for a particular organization’s business rules.

Universal data models are templates for

- one or more core subject areas such as:
  - Customers, products, accounts, documents
- and/or core business functions such as:
  - Purchasing, accounting, receiving, etc.

Introduction
Data Requirements
E-R Model
- Conceptual
- E-R Syntax
- Types
- Business Rules
- **Packaged Models**

Class Model
- Domain Model
- Relationships

Security Requirements

PACKAGED DATA MODELS

Industry-specific data models are designed to be used by organizations within specific industries.

These models are based on the premise that data model patterns for organizations are similar within a particular industry.

BENEFITS OF DATABASE PATTERNS AND PACKAGED DATA MODELS

Dramatically reduced implementation times and costs

- Provides a starting point for asking requirements questions

Higher-quality models

- Represent "best practice" data modeling techniques and data model components whose quality often exceeds that which can be achieved by internal development teams, given typical organizational pressures

BENEFITS OF DATABASE PATTERNS AND PACKAGED DATA MODELS

Functional models represent system behavior

Structural models represent system objects and their relationships:

- People
- Places
- Things

DOMAIN CLASSES

Attribute— describes one piece of information about each instance of the class

- Customer has first name, last name, phone number

Identifier or key

- One attribute uniquely identifies an instance of the class. Required for data entities, optional for domain classes. Customer ID identifies a customer

Compound attribute

- Two or more attributes combined into one structure to simplify the model. (E.g., address rather than including number, street, city, state, zip separately). Sometimes an identifier or key is a compound attribute.

Introduction
Data Requirements
E-R Model
- Conceptual
- E-R Syntax
- Types
- Business Rules
- Packaged Models

Class Model
- **Domain Model**
- Relationships

Security Requirements

ATTRIBUTES AND VALUES

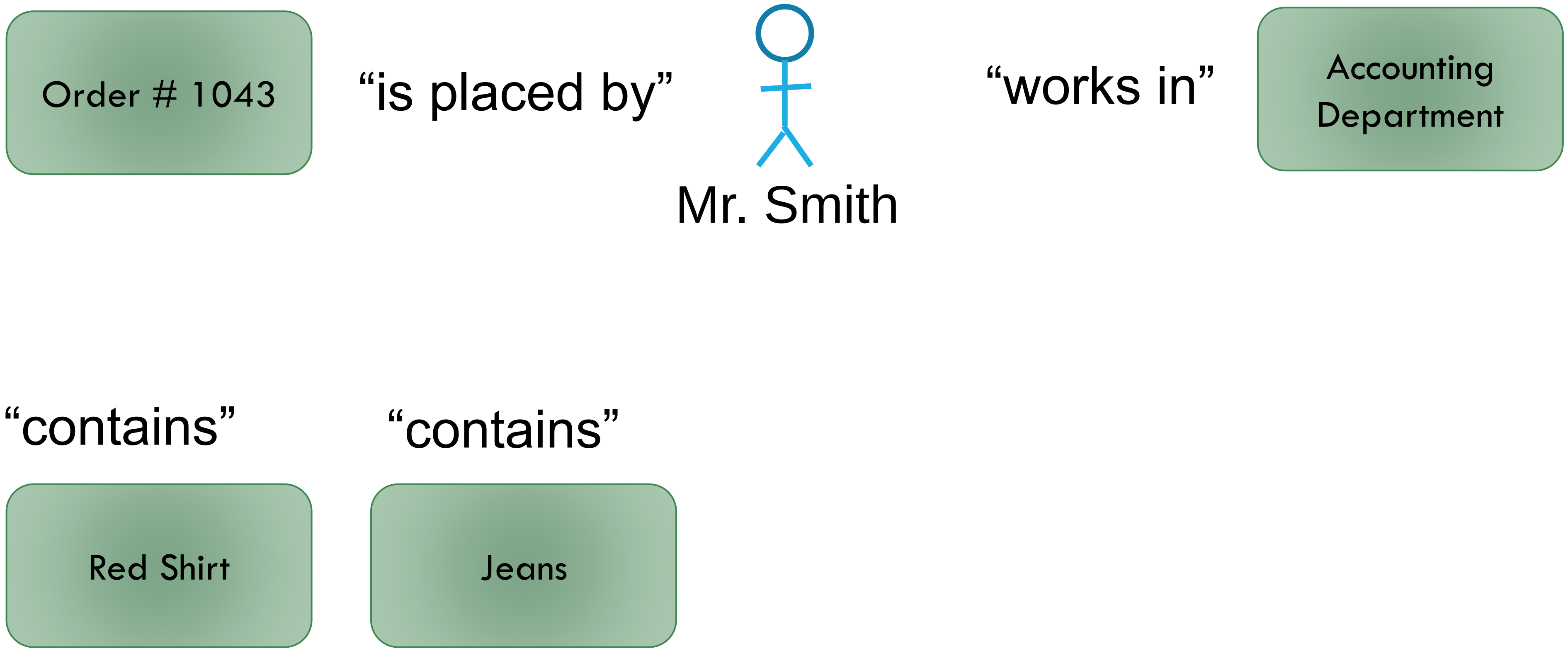
Class is a type of thing. Object is a specific instance of the class.
Each instance has its own values for an attribute.

All Customers have attributes:	**Each customer has a value for each attribute:**		
Customer ID	101	102	103
First Name	John	Dagny	Henry
Last Name	Galt	Taggart	Reardon
Home Phone	555-9182	423-1298	874-1297
Work Phone	555-3425	423-3419	874-8546

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

ASSOCIATIONS AMONG THINGS

Association– A naturally occurring relationship between classes (UML term)



Order # 1043

“is placed by”

Mr. Smith

“works in”

Accounting Department

“contains”

“contains”

Red Shirt

Jeans

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

JUST TO CLARIFY...

Called ***association*** on class diagram in UML

- Multiplicity is term for the number of associations between classes: 1 to 1 or 1 to many
- *We are emphasizing UML in this unit*

Called ***relationship*** on ERD in database class

- Cardinality is term for number of relationships in entity relationship diagrams: 1 to 1 or 1 to many

Associations and Relationships apply in two directions

- Read them separately each way
- A customer places an order
- An order is placed by a customer

Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

MINIMUM AND MAXIMUM MULTIPLICITY

Associations have minimum and maximum constraints

- Minimum is zero
  - The association is optional
- Minimum is at least one
  - The association is mandatory

Mr. jones has placed no order yet, but there might be many placed over time → Multiplicity is zero or more-optional relationship

A particular order is placed by Mr. Smith. There can’t be an order without stating the customer. → Multiplicity is one and only one-mandatory relationship

An order contains at least an item, but could have many items → Multiplicity is one or more-mandatory relationship

Introduction
Data Requirements
E-R Model
- Conceptual
- E-R Syntax
- Types
- Business Rules
- Packaged Models

Class Model
- **Domain Model**
- Relationships

Security Requirements

TYPES OF ASSOCIATIONS

Binary Association

- Associations between exactly two different classes
  - Course Section includes Students
  - Members join Club

Unary Association (recursive)

- Associations between two instances of the same class
  - Person married to person
  - Part is made using parts

Ternary Association (three)

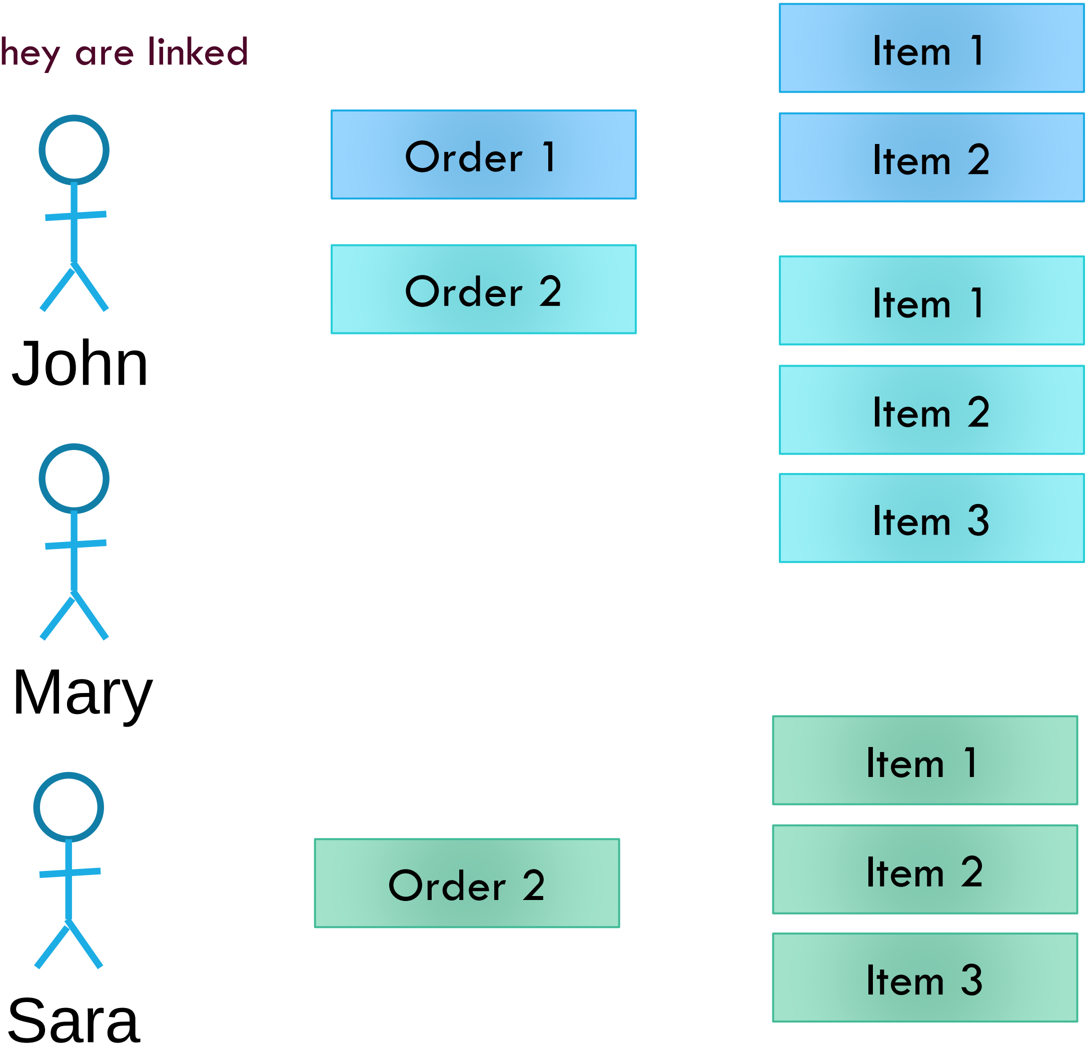
N-ary Association (between n)

Introduction
Data Requirements
E-R Model
  Conceptual
  E-R Syntax
  Types
  Business Rules
  Packaged Models
Class Model
  Domain Model
  Relationships
Security Requirements

SEMANTIC NET

Shows instances and how they are linked

- Example:



John

Order 1

Item 1

Item 2

Order 2

Item 1

Item 2

Item 3

Mary

Sara

Order 2

Item 1

Item 2

Item 3

DOMAIN CLASS MODEL DIAGRAM

Class

- A category of classification used to describe a collection of objects

Domain Class

- Classes that describe objects in the problem domain

Class Diagram

- A UML diagram that shows classes with attributes and associations (plus methods if it models software classes)

Domain Model Class Diagram

- A class diagram that only includes classes from the problem domain, not software classes so no methods

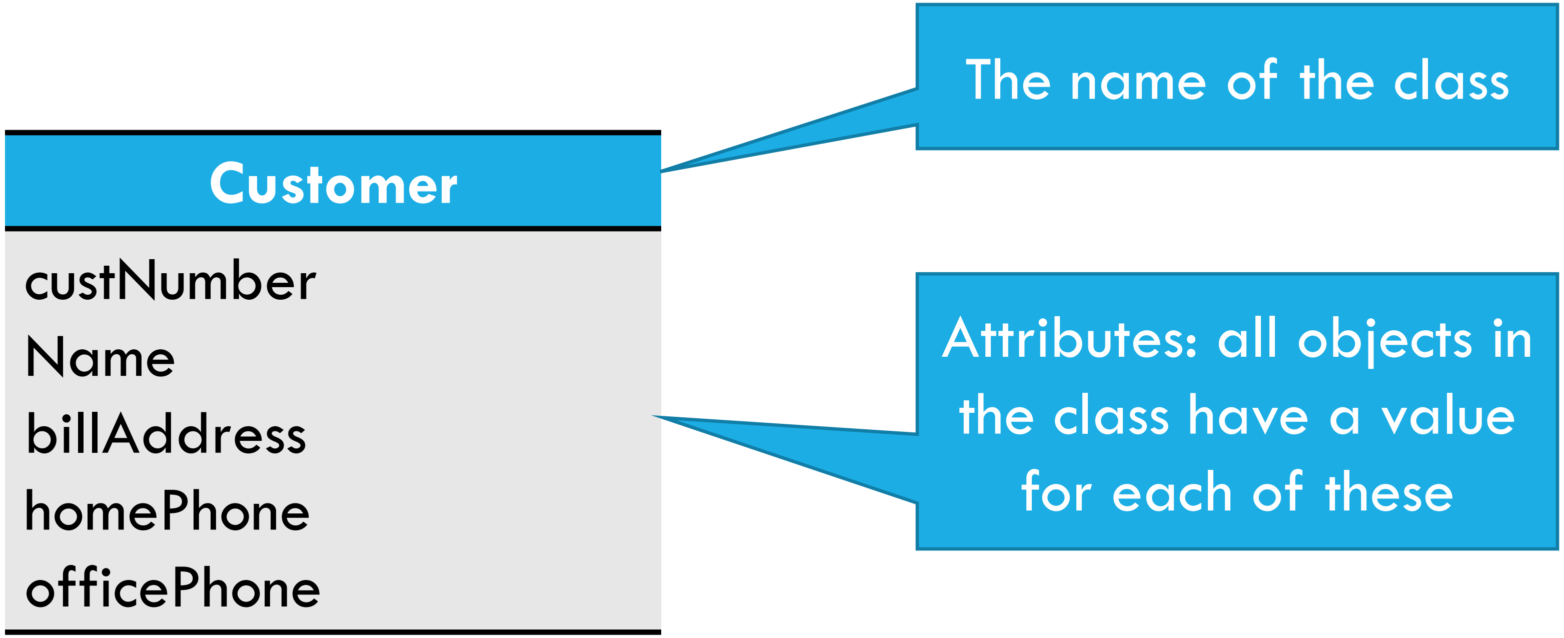
Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

DOMAIN CLASS NOTATION

Domain class has no methods

Class name is always capitalized

Attribute names are not capitalized and use camelback notation (words run together and second word is capitalized)



Customer

custNumber
Name
billAddress
homePhone
officePhone

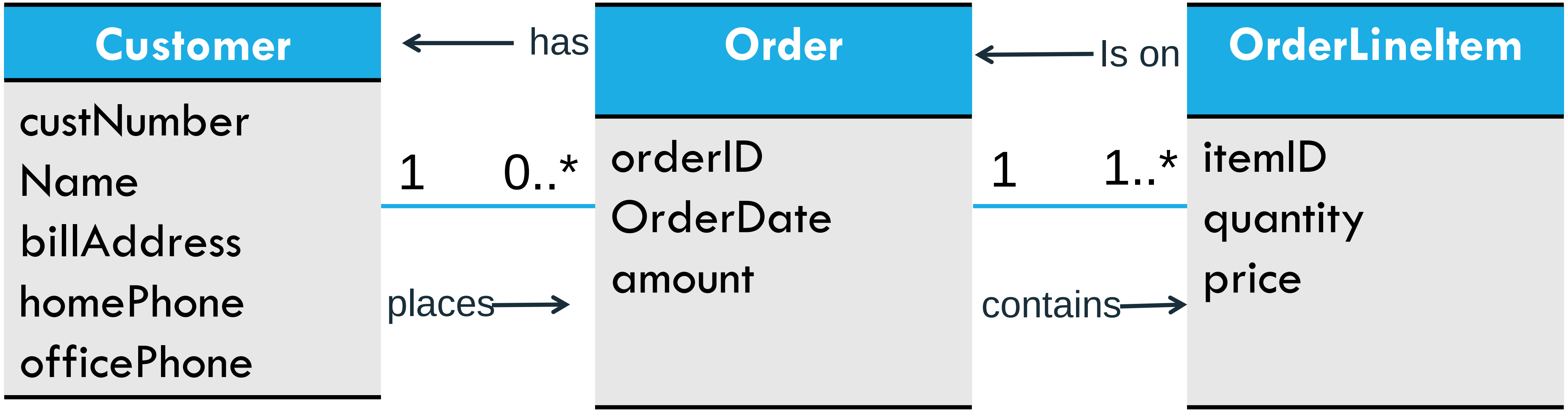
The name of the class

Attributes: all objects in the class have a value for each of these

SIMPLE DOMAIN MODEL CLASS DIAGRAM

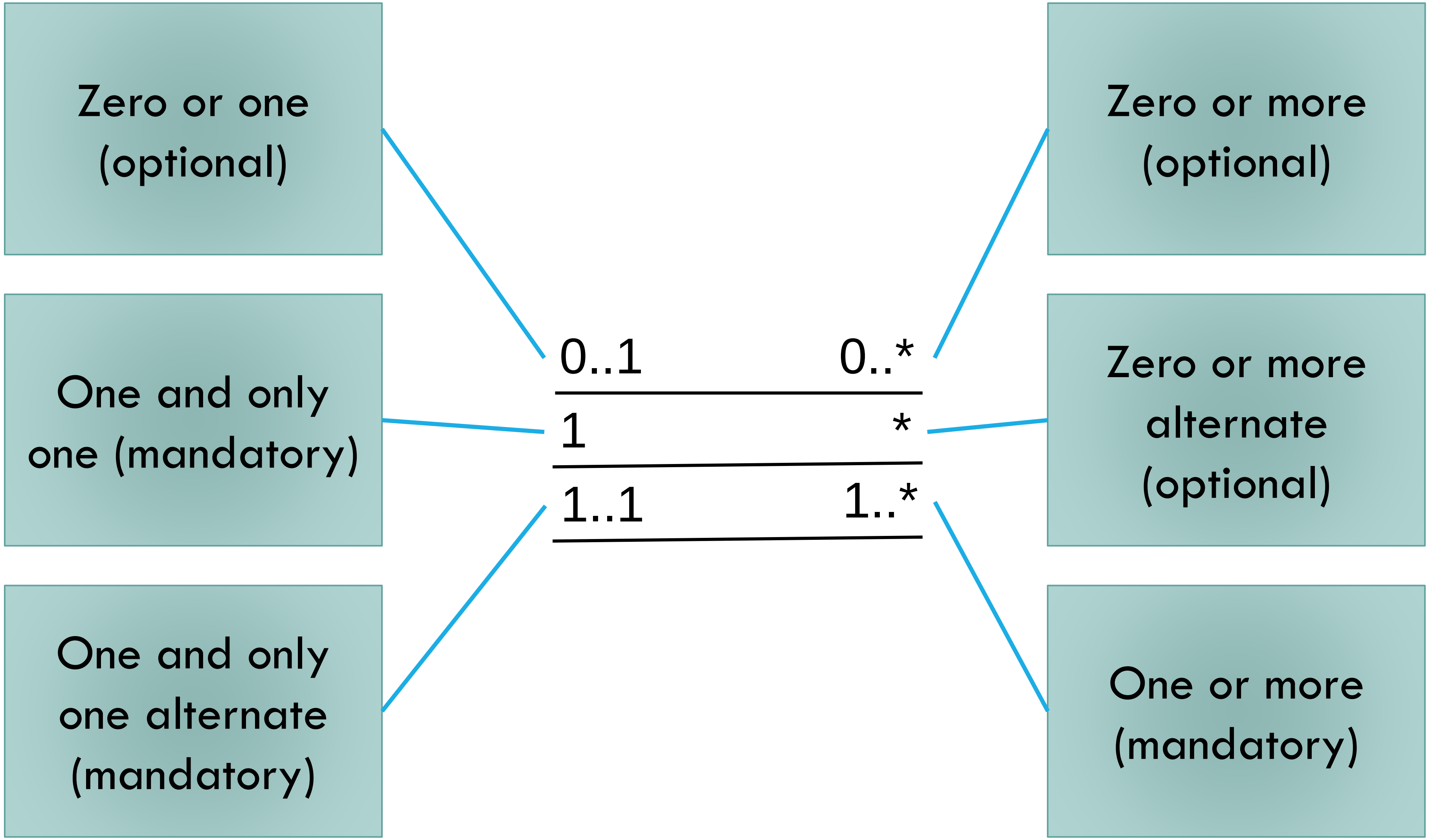
From the Semantic Net (shown previously)

- A customer places zero or more orders
- An order is placed by exactly one customer
- An order consists of one or more order items
- An order item is part of exactly one order



Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

UML NOTATION FOR MULTIPLICITY



Zero or one (optional)

One and only one (mandatory)

One and only one alternate (mandatory)

Zero or more (optional)

Zero or more alternate (optional)

One or more (mandatory)

Introduction

Data Requirements

E-R Model

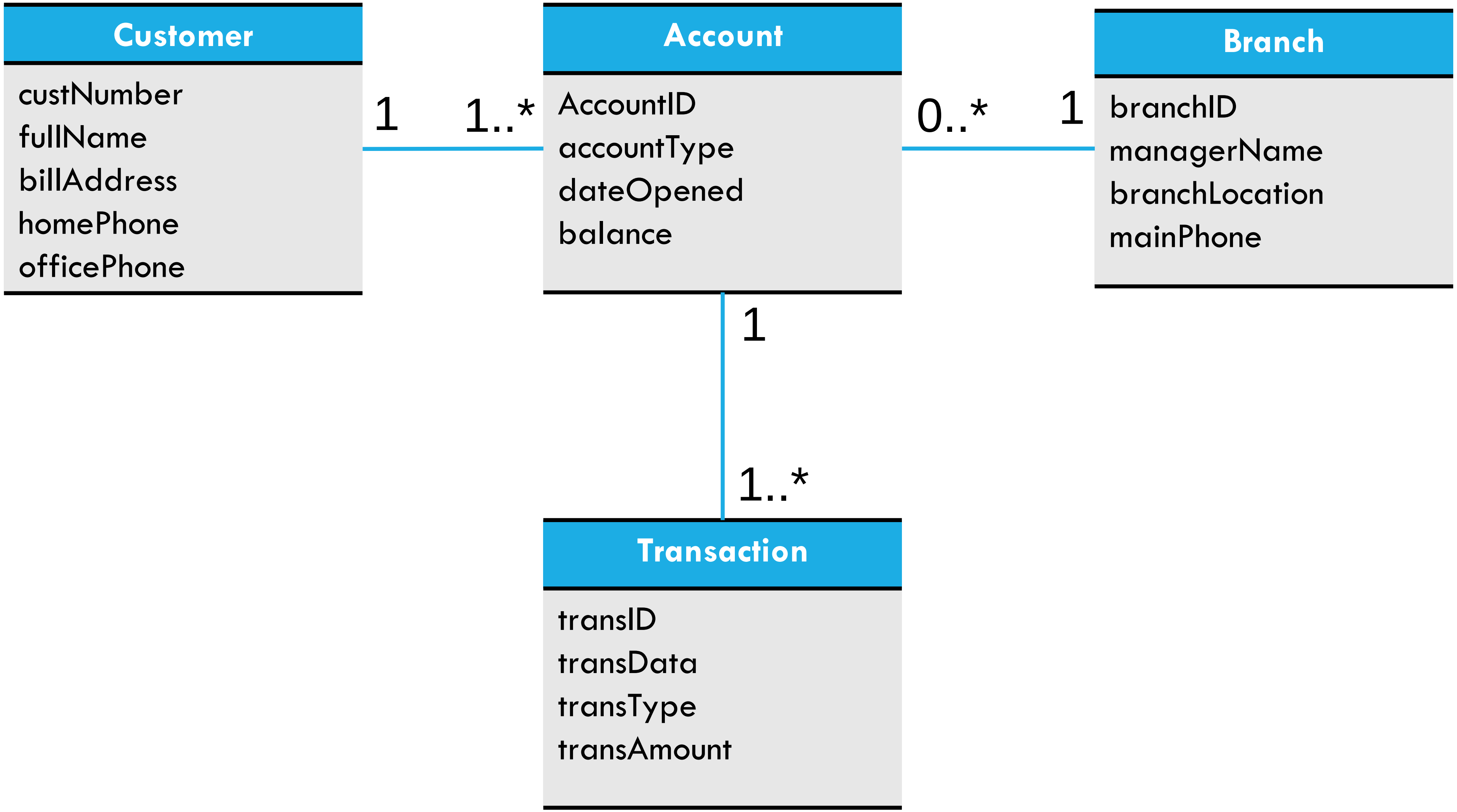
- Conceptual
- E-R Syntax
- Types
- Business Rules
- Packaged Models

Class Model

- Domain Model
- **Relationships**

Security Requirements

DOMAIN MODEL CLASS DIAGRAM



Customer

custNumber
fullName
billAddress
homePhone
officePhone

Account

AccountID
accountType
dateOpened
balance

Branch

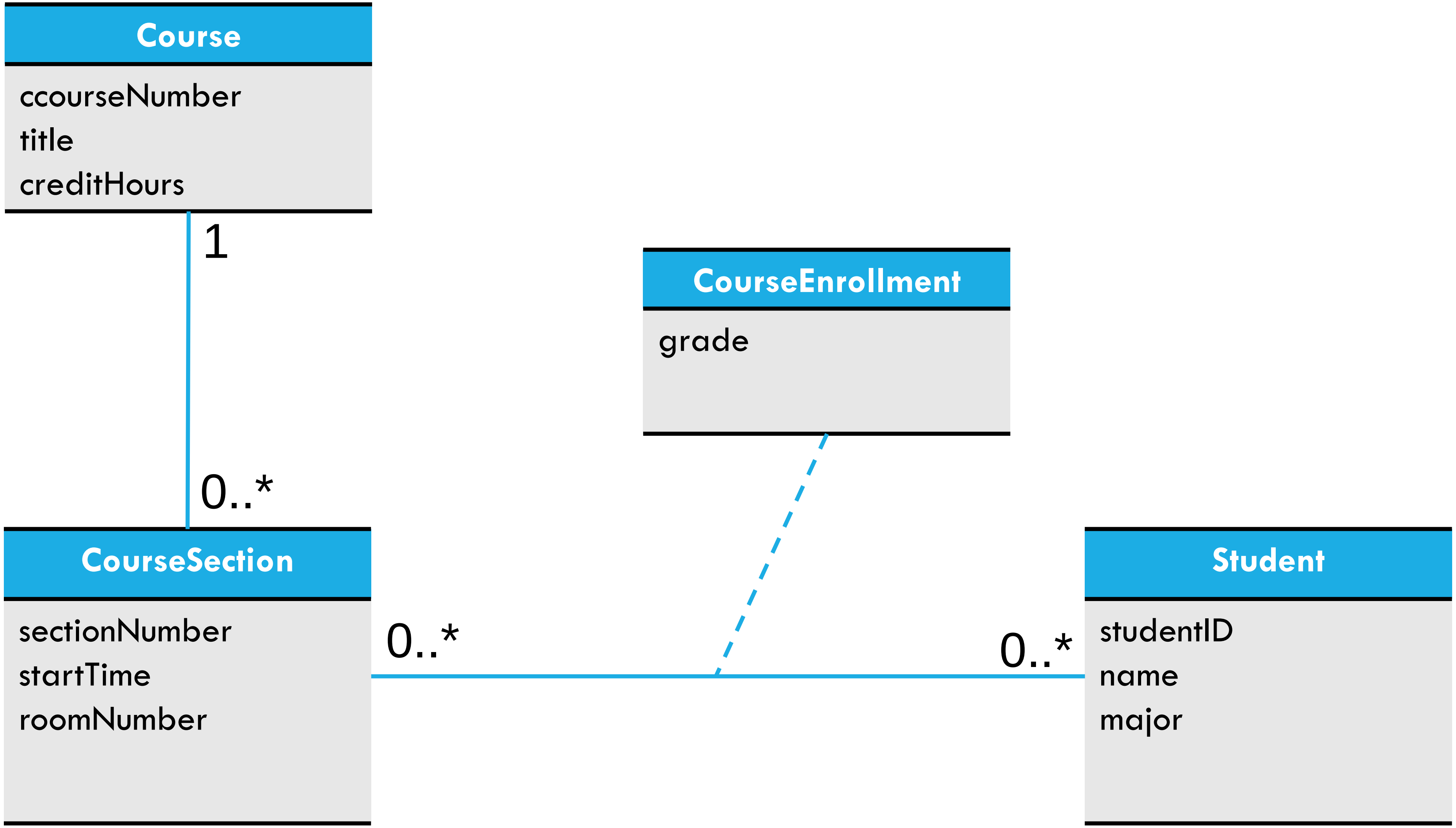
branchID
managerName
branchLocation
mainPhone

Transaction

transID
transData
transType
transAmount

Introduction
Data Requirements
E-R Model
    Conceptual
    E-R Syntax
    Types
    Business Rules
    Packaged Models
Class Model
    Domain Model
    Relationships
Security Requirements

DOMAIN MODEL CLASS DIAGRAM



Course

ccourseNumber
title
creditHours

CourseEnrollment

grade

CourseSection

sectionNumber
startTime
roomNumber

Student

studentID
name
major

GENERALIZATION AND SPECIALIZATION RELATIONSHIPS

Generalization/Specialization

- A hierarchical relationship where subordinate classes are special types of the superior classes. Often called an Inheritance Hierarchy

Superclass

- the superior or more general class in a generalization/specialization hierarchy

Subclass

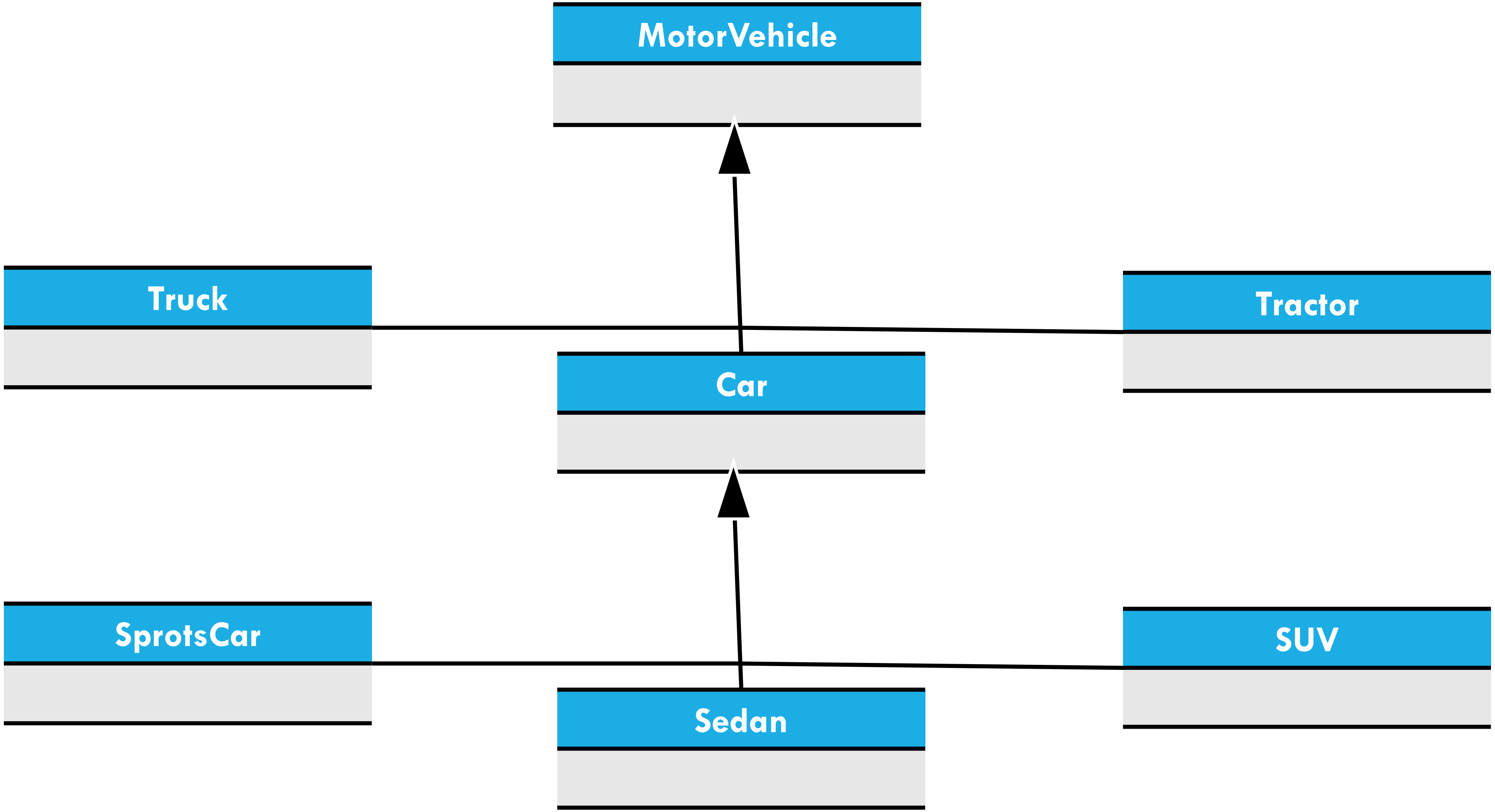
- the subordinate or more specialized class in a generalization/specialization hierarchy

Inheritance

- the concept that subclasses inherit characteristics of the more general superclass

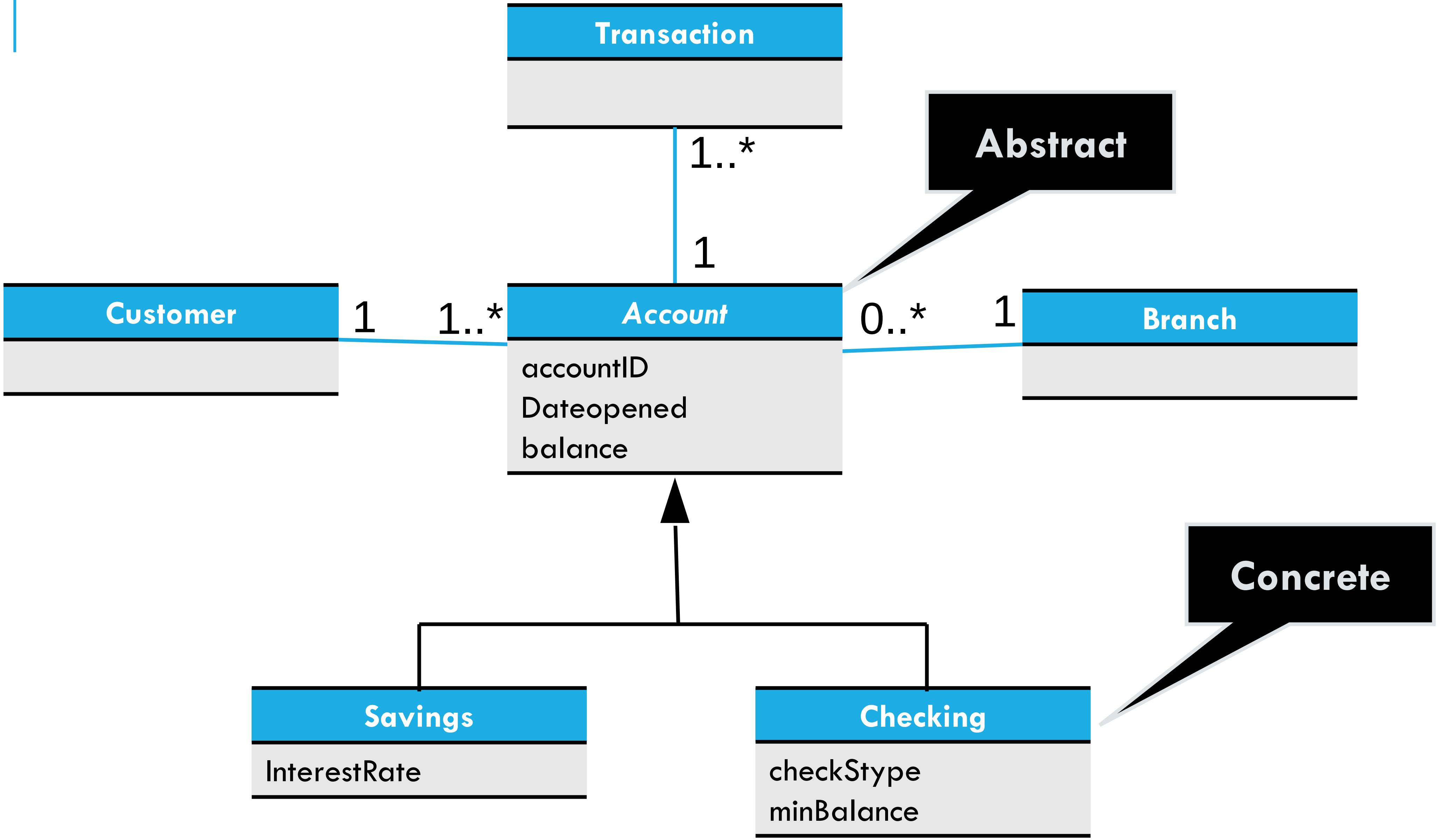
Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

GENERALIZATION AND SPECIALIZATION RELATIONSHIPS



Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

GENERALIZATION AND SPECIALIZATION RELATIONSHIPS



Introduction
Data Requirements
E-R Model
Conceptual
E-R Syntax
Types
Business Rules
Packaged Models
Class Model
Domain Model
Relationships
Security Requirements

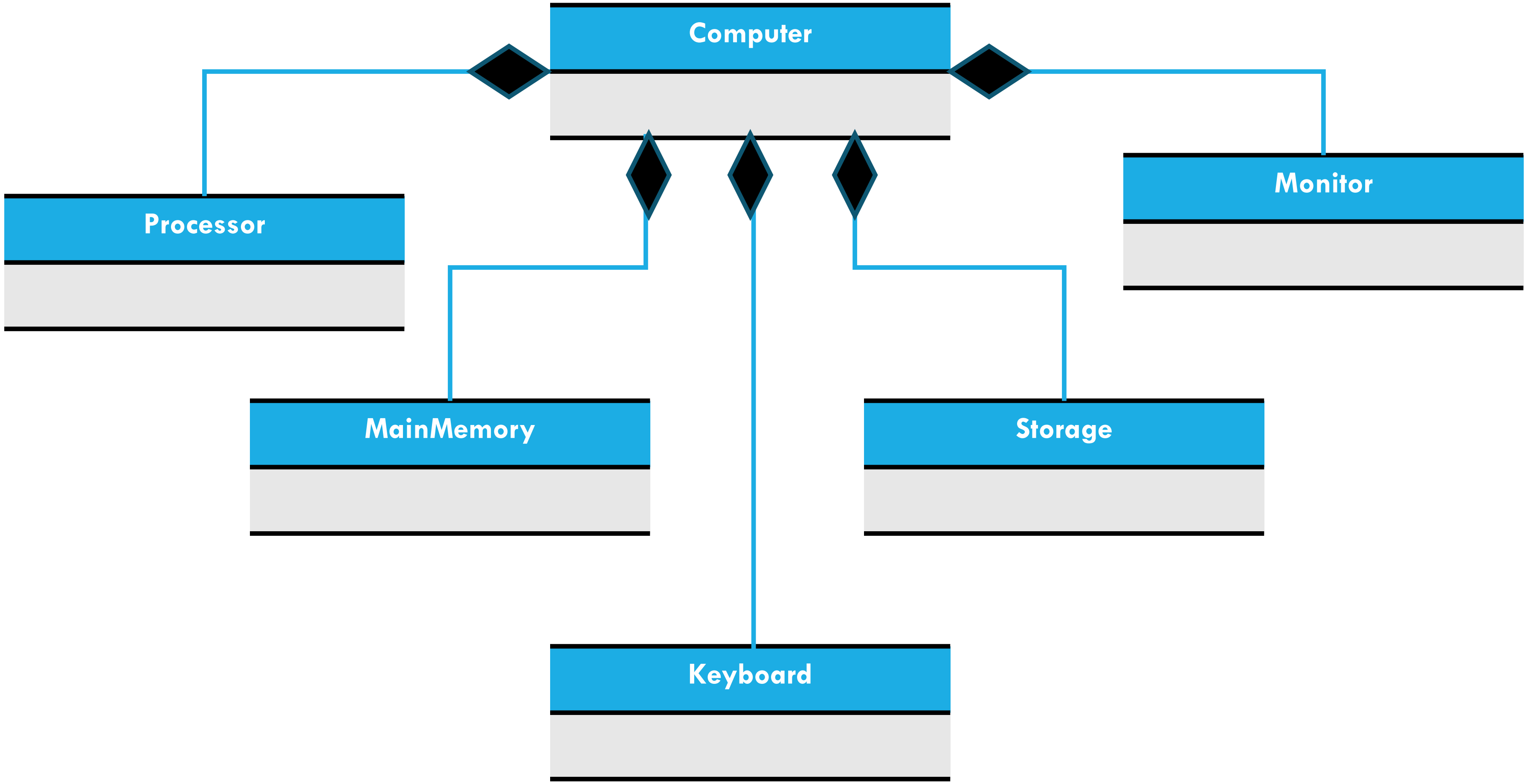
WHOLE PART RELATIONSHIPS

Whole-part relationship— a relationship between classes where one class is part of or a component portion of another class

- Aggregation— a whole part relationship where the component part exists separately and can be removed and replaced (UML diamond symbol, next slide)
  - Computer has disk storage devices
  - Car has wheels
- Composition— a whole part relationship where the parts can no longer be removed (filled in diamond symbol)
  - Hand has fingers
  - Chip has circuits

Introduction
Data Requirements
E-R Model
  Conceptual
  E-R Syntax
  Types
  Business Rules
  Packaged Models
Class Model
  Domain Model
  Relationships
Security Requirements

WHOLE PART RELATIONSHIPS



ONE FINAL NOTE ON RELATIONSHIPS…

There are actually three types of relationships in class diagrams

- Association Relationships
  - These are associations discussed previously, just like ERD *relationships*
- Whole Part Relationships
  - One class is a component or part of another class
- Generalizations/Specialization Relationships
  - Inheritance

Try not to confuse relationship with association