King Saud University

College of Engineering

IE – 462: "Industrial Information Systems"

Spring – 2019 (2nd Sem. 1439-40H)

Chapter 3

Data Modeling and Design – p2 – E-R Diagram

Prepared by: Ahmed M. El-Sherbeeny, PhD

Lesson Overview

- Introduction – (p1)
- **E-R Diagram** – (p2)
- Case Studies – (p3)

Lesson Overview

- **E-R Diagram** – (part-i)
  - **Introduction to E-R Modeling**
    - Introduction
    - Entities
    - Attributes
    - Candidate Keys and Identifiers
    - Other Attribute Types
    - Relationships

Lesson Overview

- **E-R Diagram** – (part-ii)
  - **Conceptual Data Modeling and the E-R Model**
    - Degree of a Relationship
      - Unary Relationships
      - Binary Relationships
      - Ternary Relationships
    - Cardinalities in Relationships
      - Minimum and Maximum Cardinalities
    - Naming and Defining Relationships
    - Associative Entities
    - Summary

Lesson Overview

- **E-R Diagram** – (part-ii – cont.)
  - **Supertypes and Subtypes**
  - **Business Rules**

INTRODUCTION TO E-R MODELING

Introduction

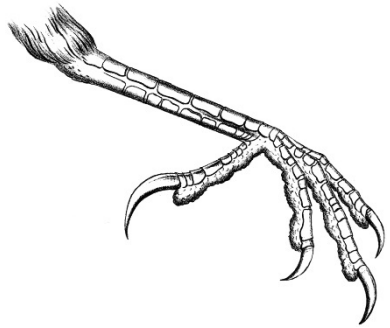
Introduction to E-R Modeling

- Purpose of E-R modeling is to design a conceptual schema (model) of entities *and* their relationships for an organization/business
- **Entity-relationship data model** (E-R model): detailed, logical representation of:
  - **data entities**
  - **relationships**, and
  - **attributes**: they represent properties of *both* the entities and their relationships

Introduction to E-R Modeling

Entity-relationship diagram (ERD):

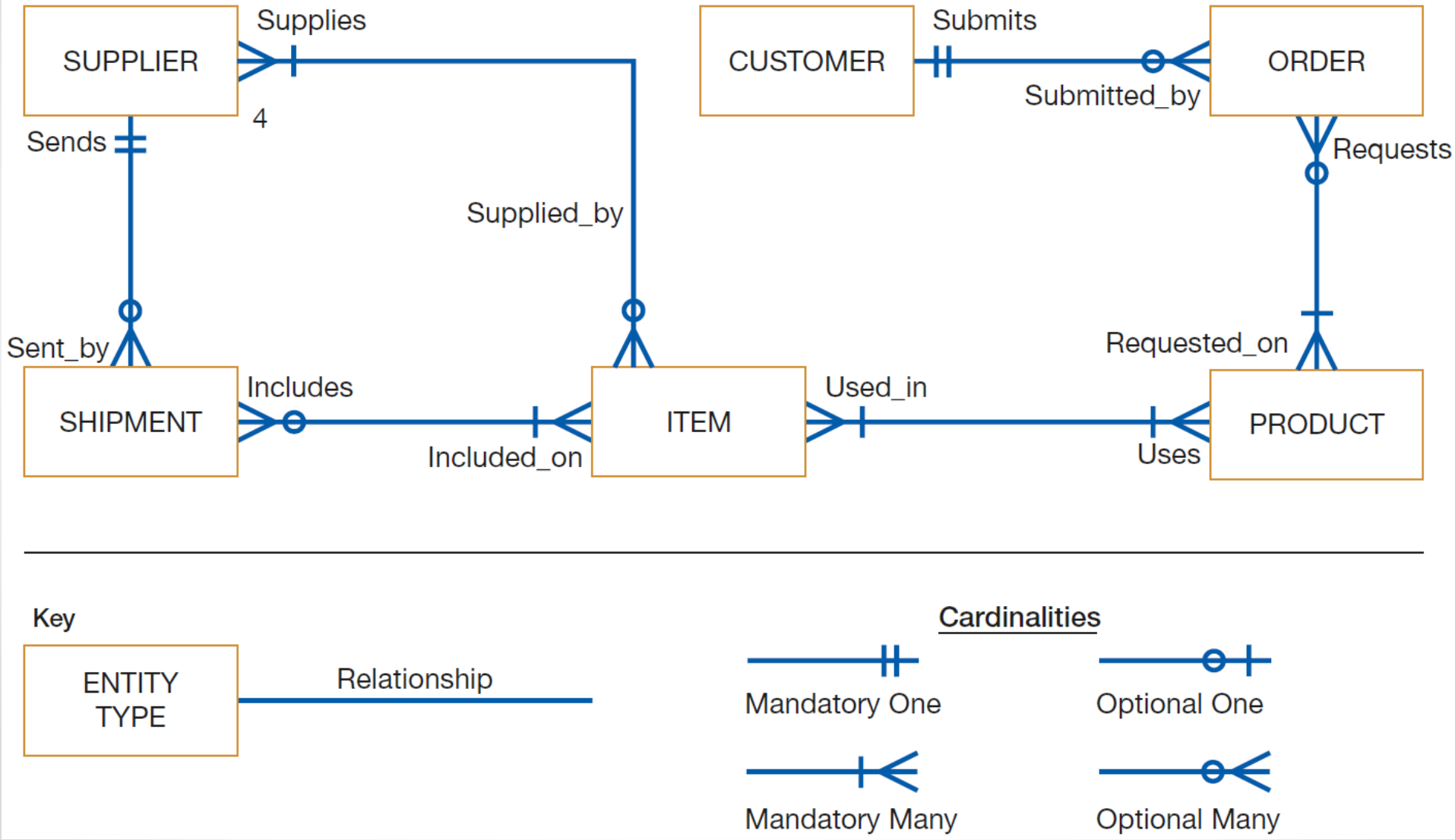
- graphical representation of an E-R model
- utilizes several notations to show data in terms of the entities and relationships described by that data
  - notation mostly uses "**crow's foot**" symbols



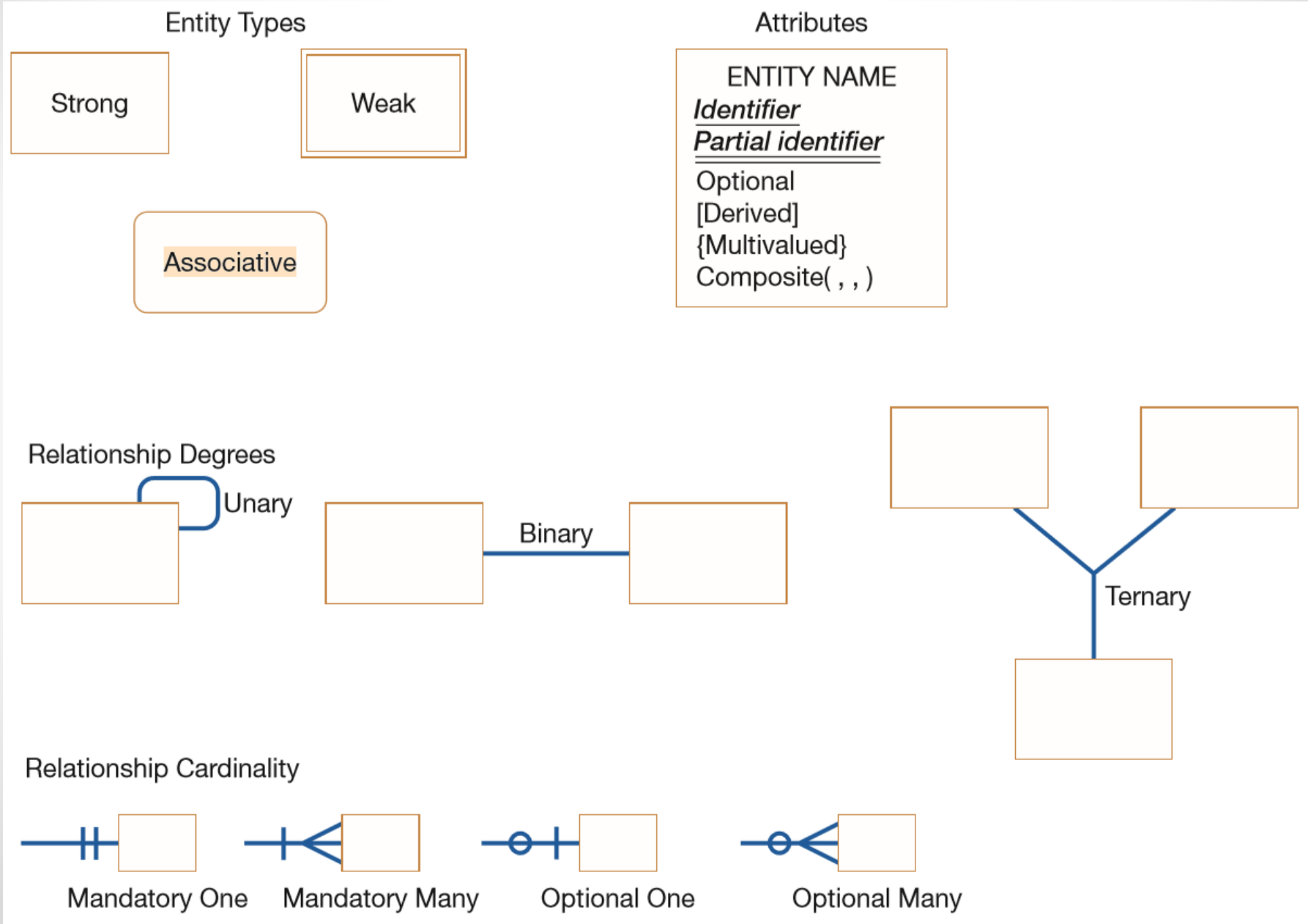
- places data attribute names within entity rectangles
- see next 2 slides for notations, which will be explained in detail in following sections

EMPLOYEE
Employee_ID
Employee_Name(. . .)
Birth_Date

Sample E-R Diagram



Basic E-R Notation



Entities

Entities

Entity:

- Class of *persons*, *places*, *objects*, *events*, or *concepts* for which the organization wishes to maintain data
- Represented by *Q1* in Table 8-1
- Each entity must have a *unique identity* that distinguishes it from each other entity

Entities

Examples of Entities (cont.):

- **Persons**: agency, contractor, customer, department, division, employee, instructor, student, supplier
- **Places**: sales region, building, room, branch office, campus, store, warehouse, state, shop floor
- **Objects**: book, machine, part, product, raw material, software license, software package, tool, vehicle model
- **Events**: application, award, cancellation, class, flight, order, registration, renewal, requisition, reservation, sale, trip, assignment
- **Concepts**: account, block of time, bond, course, fund, qualification, stock, work center

Entities

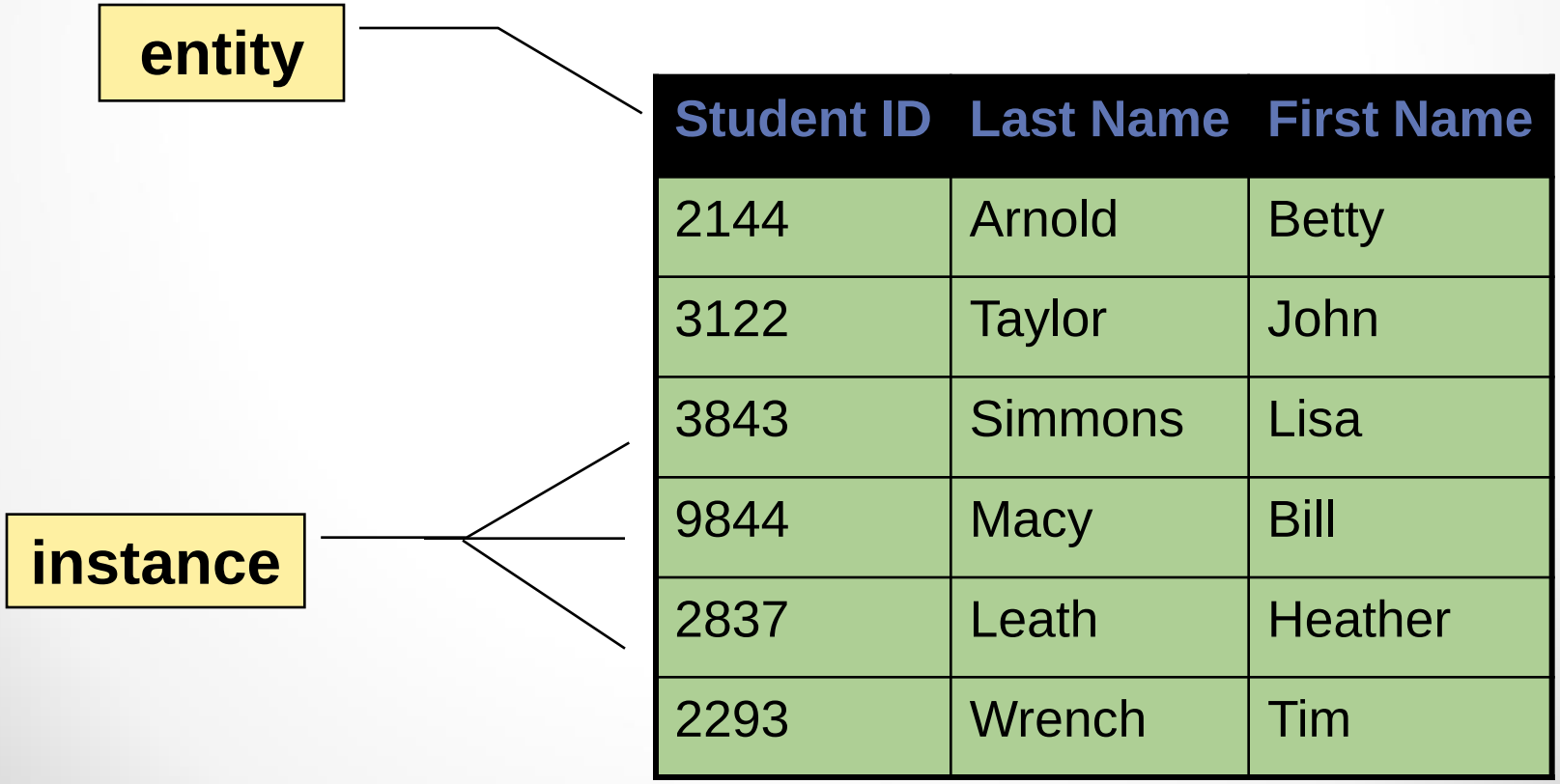
Entity *Types* vs. Entity *Instances*:

- Important to distinguish between *entity types* and *entity instances*
- **Entity type** (aka **entity class** or –simply– **entity**):
  - collection of entities that share common properties/characteristics
  - *each entity* type in an E-R model is given a name
  - name is placed inside a rectangle representing the entity
  - each entity is described just *once* in a data model

Entities

Entity *Types* vs. Entity *Instances* (cont.):

- **Entity instance** (aka **instance**):
  - a single occurrence of an entity
  - *many* instances of an entity type may be represented by data stored in the database



Entities

Common Mistakes with Data Entities:

- Many people confuse
  - data *entities* with *sources/sinks* or system outputs,
  - *relationships* with *data flows*
- Avoid this problem with a simple rule:
  - true data entity will have *many possible instances*,
  - each instance has a distinguishing characteristic
- Example below, “sorority expense system”:
  - do we need to keep track of data about the treasurer?

Entities

Naming Entity Types:

- Should use all *capital letters*
  - e.g. EMPLOYEE
- Should be named by a *singular noun*
  - e.g. CUSTOMER, STUDENT, or AUTOMOBILE
- Use *simple*, *concise* nouns
  - e.g. use REGISTRATION instead of STUDENT REGISTRATION FOR CLASS
- Name should be *descriptive/specific* to company
  - e.g. instead of just using ORDER, use PURCHASE ORDER (to distinguish between it and CUSTOMER ORDER)

Attributes

Attributes

Attribute:

- A named *property* or *characteristic* of an entity that is of interest to the organization
- Represented by *Q3* in Table 8-1
- Some typical entity types and associated attributes:
  - STUDENT: Student_ID, Student_Name, Home_Address, Phone_Number, Major
  - AUTOMOBILE: Vehicle_ID, Color, Weight, Horsepower
  - EMPLOYEE: Employee_ID, Employee_Name, Payroll_Address, Skill

Attributes

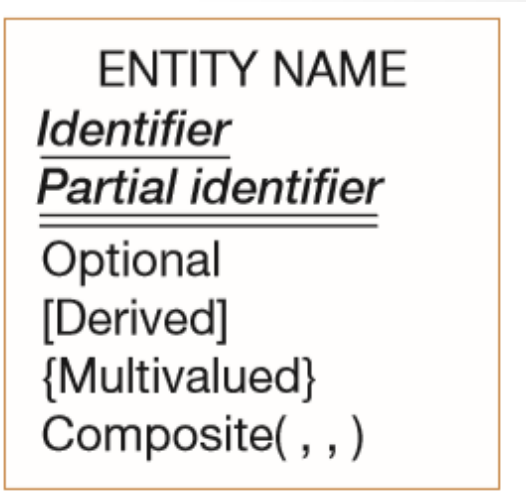
Naming Attributes:

- Use initial capital letter, followed by lowercase letters
- Use nouns for names; e.g. Age
- Use underscores to separate words (optional); e.g. Customer_ID, Product_Minimum_Price
- Attribute name should be *unique*:
  - no 2 attributes of same entity type may have the same name
  - preferable no two attributes have the same name (i.e. across all entity types)
- Follow a standard format for naming attributes:
  - e.g. using Student_GPA as opposed to GPA_of_Student

Attributes

Using Attributes in E-R Diagram:

- Place the name inside the rectangle for associated entity
- We use different notations to distinguish between different types of attributes (to be discussed next)



Candidate Keys and Identifiers

Candidate Keys and Identifiers

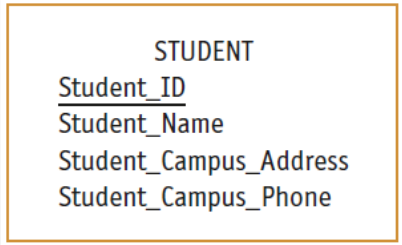
Candidate Key (aka **Primary key**):

- It's an attribute (or combination of attributes) that *uniquely* identifies each instance of an entity type
- Represented by *Q2* in Table 8-1
- e.g. candidate key for a STUDENT entity type might be Student_ID

Candidate Keys and Identifiers

Identifiers:

- Some entities may have > 1 possible candidate key; e.g. for EMPLOYEE data entity:
  - possible candidate key: Employee_ID
  - another possible candidate key: Employee_Name and Address (assuming no two employees with the same name live at the same address)
  - designer must choose one of the candidate keys as identifier thus:
- Identifier: candidate key that has been selected as the *unique*, *identifying characteristic* for entity type
  - it is represented by placing a *solid underline* below identifier



Candidate Keys and Identifiers

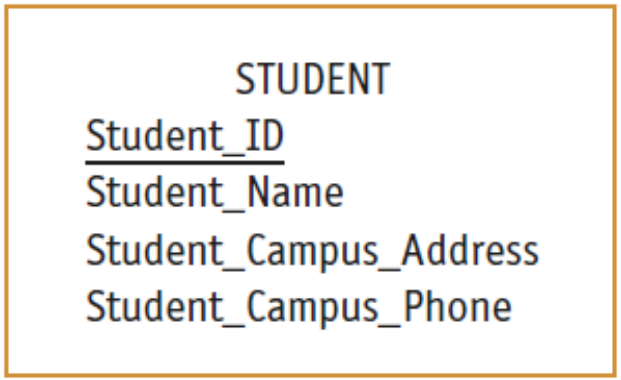
Criteria for Selecting Identifiers:

- Choose a candidate key that will *not change* its value over life of each instance of the entity type
  - e.g. don't pick identifier for EMPLOYEE: combination of Employee_Name and Payroll_Address
- Choose candidate key so that, for each instance of the entity, the attribute is guaranteed to have:
  - valid values and
  - not be 'null' (note, special controls in data entry can eliminate possibility of errors, e.g. use of '*')

Candidate Keys and Identifiers

Criteria for Selecting Identifiers (cont.):

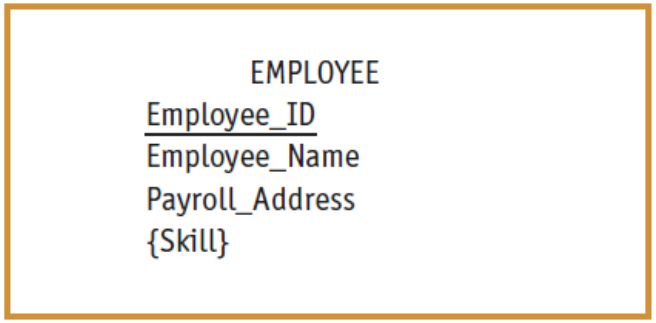
- Avoid so-called "intelligent identifiers"
  - e.g. first 2 digits of a key for a PART entity may indicate the warehouse location
  - note that such codes are often modified, and this would make primary key values invalid
- Example here:
  - representation for a STUDENT entity type using E-R notation
  - STUDENT has:
    - a simple identifier, Student_ID, and
    - 3 other simple attributes



Other Attribute Types

Other Attribute Types

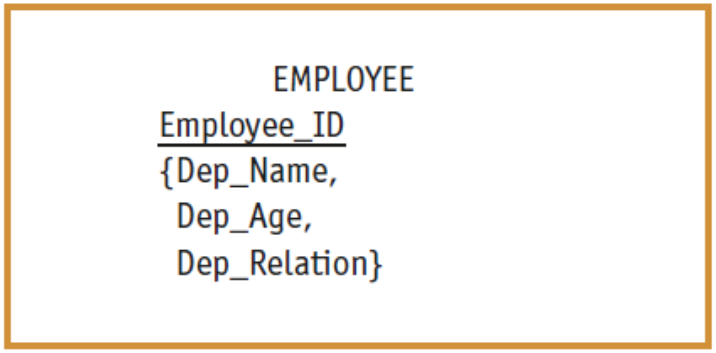
- **Single-valued attribute:**
  - attribute that may take one entry in each instance of that attribute
  - e.g. there is only 1 employee ID number to be entered in each instance of the attribute Employee_ID
- **Multi-valued attribute**:
  - an attribute that may take on > 1 value for each entity instance
  - e.g. Skill is a multivalued attribute (since each employee can have > 1 skill)
  - special symbol indicates that it is multivalued: { }



Other Attribute Types

Repeating group:

- A set of two or more multi-valued attributes that are logically related
- e.g. employee entity with multivalued attributes for data about each employee's dependents:
  - data includes: dependent name, age, and relation to employee
  - dependents: spouse, child, parent, etc.
  - data are multivalued attributes about employee
  - we show this by using one set of curly brackets around the data that repeats together
  - we call this a *repeating group*



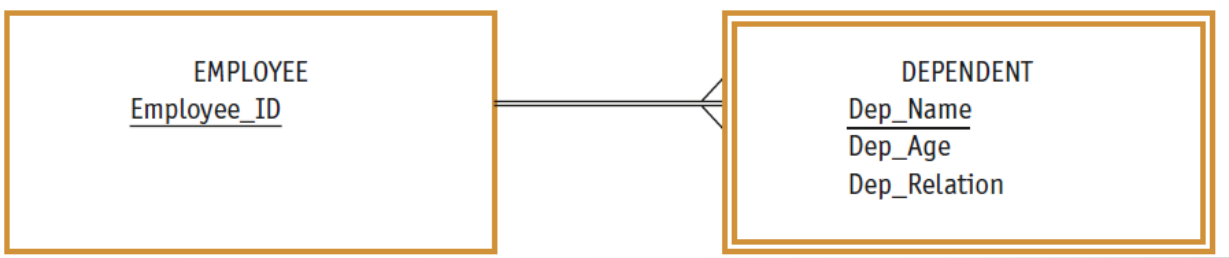
Repeating group of dependent data

Other Attribute Types

Weak entity:

- Second approach to representing a repeating group:
  - consider dependents as *entities*
  - we separate the repeating data into another entity, called a *weak* (or *attributive*) *entity*
  - weak entity is designated by:
    - *rectangle* with a *double line border* and
    - relationship to link the weak entity to its associated regular entity (using *double line*)

Weak entity for dependent data

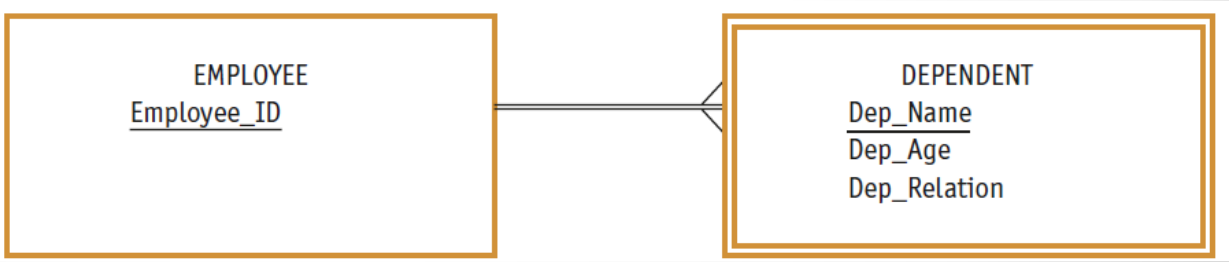


Other Attribute Types

Weak entity (cont.):

- Examine example below:
  - use a weak entity, DEPENDENT
  - establish relationship between DEPENDENT and EMPLOYEE
  - crow's foot next to DEPENDENT: there may be many DEPENDENTs for the same EMPLOYEE
  - identifier of DEPENDENT: dependent's name + ID of the employee, or use a double underline for Dep_Name to designate it as a *partial identifier*

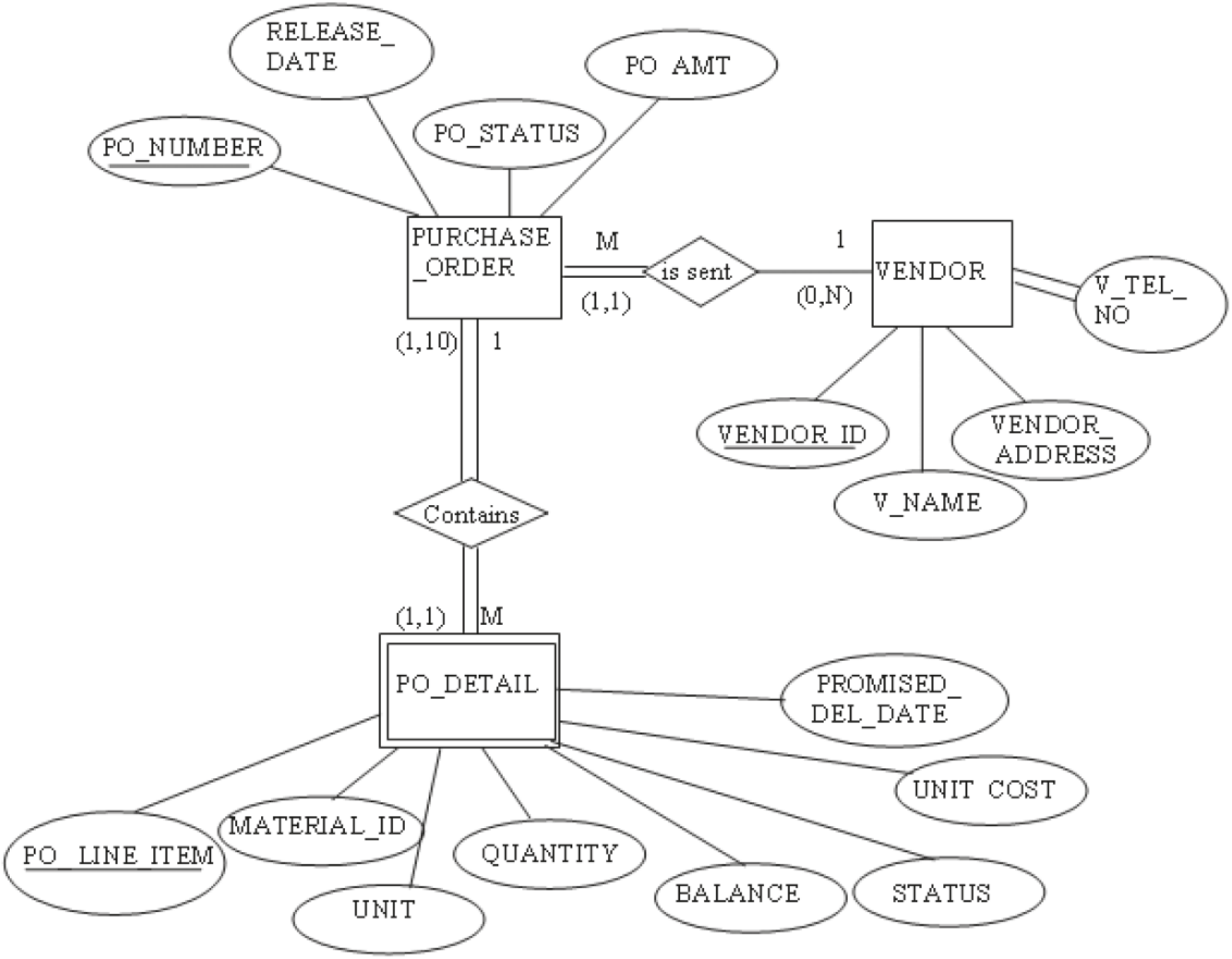
Weak entity for dependent data



Weak and Strong Entities

- E-R diagram uses a box to represent an entity set (PURCHASE_ORDER, PO_DETAIL, and VENDOR)
- E-R diagrams distinguish between *weak* and *strong* entities
  - entity is *weak* if its existence is dependent on the existence of another entity
  - e.g. of this occurs in the case of PO_DETAIL: PO_DETAIL is dependent on the existence of PURCHASE_ORDER

Weak and Strong Entities



Other Attribute Types

- **Required attribute**: an attribute that *must* have a value for every entity instance
  - shown in bold letters (list!)
- **Optional attribute**: an attribute that *may not* have a value for every entity instance
  - shown in normal letters (list!)

Required, optional, composite, and derived attributes

Other Attribute Types

- **Composite attribute**: an attribute that has meaningful *component* parts
  - e.g. Name or Address
  - components are shown between brackets ( ) (list!)
- **Derived attribute**: an attribute whose value can be computed from *related attribute* values
  - shown inside square brackets: [ ] (list!)
- Notations for each attributes type is shown below

ENTITY NAME
Identifier
Partial identifier
Optional
[Derived]
{Multivalued}
Composite(, ,)

EMPLOYEE
Employee_ID
Employee_Name(First_Name, Last_Name)
Date_of_Birth
[Employee_Age]

Required, optional, composite, and derived attributes

Relationships

Relationships

Relationship:

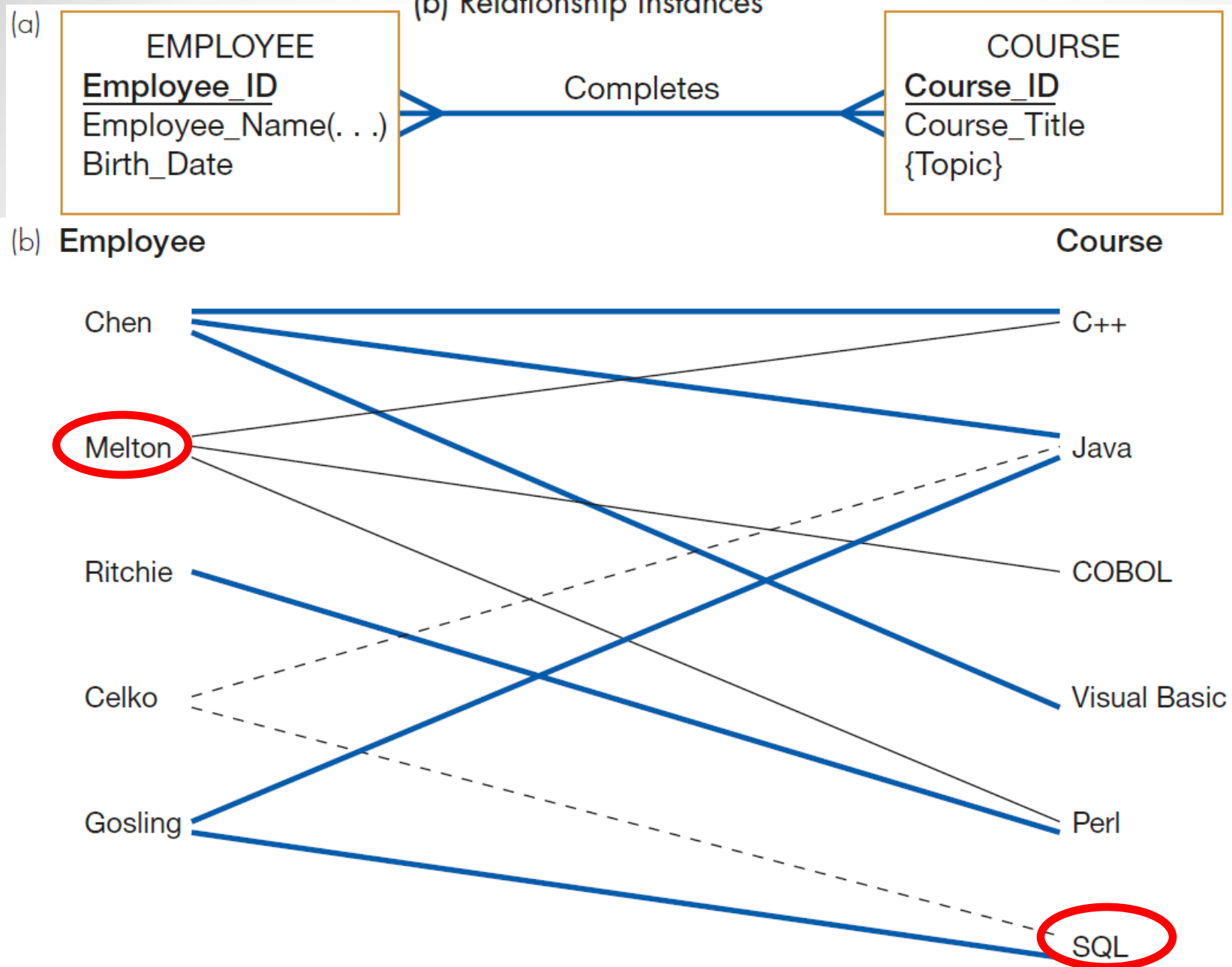
- It is association between the instances of one or more entity types that is of interest to the organization
- It is the 'glue' that holds together the various components of an E-R model
- Represented by *Questions 5, 7, and 8* in Table 8-1
- Labeled with verb phrases since usually means that an event has occurred
- Note, some standards use *two verb phrases* for a relationship name (so it can be read in two directions), while some only use *one verb phrase*

Relationships

Relationship (cont.):

- Consider example shown on next slide:
  - training department in a company wants to track which training courses employees have completed
  - this leads to a relationship called Completes between the EMPLOYEE and COURSE entity types
  - this is a *many-to-many* relationship:
    - each *employee* may complete >1 *course*
    - each *course* may be completed by >1 *employee*
  - we can use Completes relationship to determine:
    - courses a given employee has completed
    - identity of each employee who has completed a particular course

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



Videos to Watch

- **Entity Relationship Diagram (ERD) Tutorial - Part 1**
  https://youtu.be/QpdhBUYk7Kk
- **Entity Relationship Diagram (ERD) Tutorial - Part 2**
  https://youtu.be/-CuY5ADwn24
- **Entity-Relationship Diagrams** (another system)
  https://youtu.be/c0_9Y8QAstg
- **Entity Relationship Diagram (ERD) Training Video**
  https://youtu.be/-fQ-bRllhXc

Sources

- "**Chapter 3: Database Modeling and Design**"; Slides by Dr. Sabeur Kosantini (2017)
- "**Types of Database Management Systems**" (2017) by Arjun Panwar, c-sharpcorner.com; Available at: https://www.c/sharpcorner.com/UploadFile/65fc13/types-of-database-management-systems/
- **Modern Systems Analysis and Design**. Joseph S. Valacich and Joey F. George. Pearson. Eighth Ed. 2017. Chapter 8.
- **Design of Industrial Information Systems**. Thomas Boucher, and Ali Yalcin. Academic Press. First Ed. 2006. Chapter 3.

Gathering Info. for Conceptual Data Modeling

TABLE 8-1 Requirements Determination Questions for Data Modeling

1. *What are the subjects/objects of the business?* What types of people, places, things, materials, events, etc. are used or interact in this business, about which data must be maintained? How many instances of each object might exist?—**data entities and their descriptions**
2. *What unique characteristic (or characteristics) distinguishes each object from other objects of the same type?* Might this distinguishing feature change over time or is it permanent? Might this characteristic of an object be missing even though we know the object exists?—**primary key**
3. *What characteristics describe each object?* On what basis are objects referenced, selected, qualified, sorted, and categorized? What must we know about each object in order to run the business?—**attributes and secondary keys**
4. *How do you use these data?* That is, are you the source of the data for the organization, do you refer to the data, do you modify it, and do you destroy it? Who is not permitted to use these data? Who is responsible for establishing legitimate values for these data?—**security controls and understanding who really knows the meaning of data**

Gathering Info. for Conceptual Data Modeling

TABLE 8-1 Requirements Determination Questions for Data Modeling

5. *Over what period of time are you interested in these data?* Do you need historical trends, current "snapshot" values, and/or estimates or projections? If a characteristic of an object changes over time, must you know the obsolete values?—**cardinality and time dimensions of data**
6. *Are all instances of each object the same?* That is, are there special kinds of each object that are described or handled differently by the organization? Are some objects summaries or combinations of more detailed objects?—**supertypes, subtypes, and aggregations**
7. *What events occur that imply associations among various objects?* What natural activities or transactions of the business involve handling data about several objects of the same or a different type?—**relationships and their cardinality and degree**
8. *Is each activity or event always handled the same way or are there special circumstances?* Can an event occur with only some of the associated objects, or must all objects be involved? Can the associations between objects change over time (for example, employees change departments)? Are values for data characteristics limited in any way?—**integrity rules, minimum and maximum cardinality, time dimensions of data**