Chapter 6

Data Modeling

SYSTEMS ANALYSIS AND DESIGN

SEVENTH EDITION

DENNIS, WIXOM, AND ROTH

Learning Objectives

- Explain the rules and style guidelines for creating entity relationship diagrams (ERDs).
- Create an ERD.
- Describe the use of a data dictionary and metadata.
- Explain how to balance ERDs and data flow diagrams.
- Describe the process of normalization.

Key Definitions (1 of 2)

- Data model
  - A formal way of representing the data that are used and created by a business system
  - Shows the people, places and things about which data is captured and the relationships among them.
  - *Logical* data model shows the organization of data without indicating how it is stored, created, or manipulated
  - *Physical* data model shows how the data will actually be stored in databases or files.

Key Definitions (2 of 2)

- Entity Relationship Diagram (ERD)
  - A popular way to depict the data model
- *Normalization* is the process analysts use to validate data models.
- Data models should *balance* with process models

Why Is Data Modeling Crucial?

- Data is a *resource* to be shared by as many processes as possible.
- Data organization must be flexible and adaptable to unanticipated business requirements – and *that is the purpose of data modeling.*

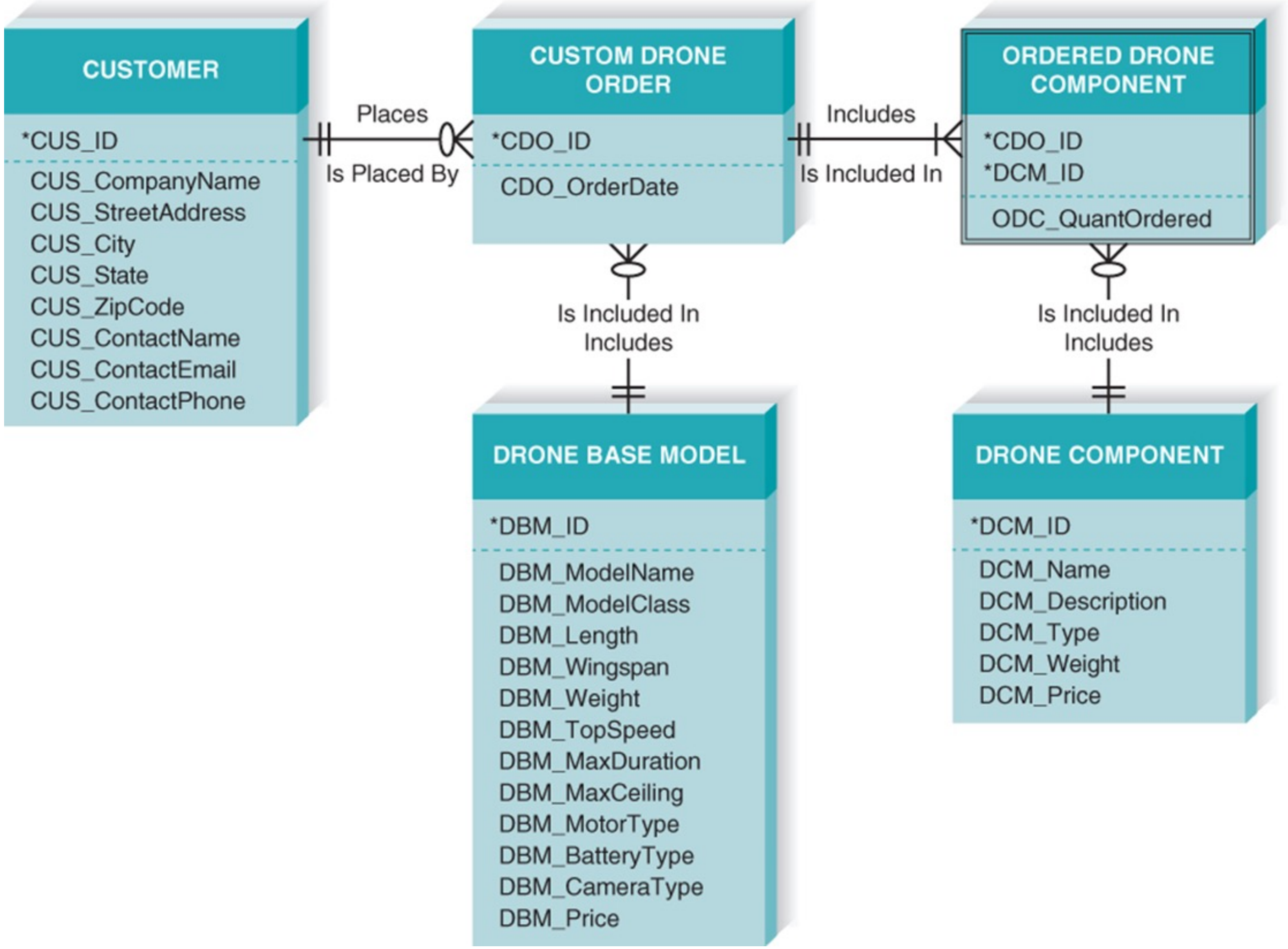
Other Data Modeling Issues…

- Data structures and properties are reasonably permanent –more stable than the processes that use the data.
- Typically very similar to the existing system.
- Data models are much smaller than process models and are constructed more rapidly.
- Constructing the data model helps analysts and users quickly reach consensus on business terminology and rules.

Entity Relationship Diagrams

WHAT DO ERDS TELL US?

Reading an ERD



CUSTOMER

*CUS_ID

CUS_CompanyName
CUS_StreetAddress
CUS_City
CUS_State
CUS_ZipCode
CUS_ContactName
CUS_ContactEmail
CUS_ContactPhone

CUSTOM DRONE ORDER

*CDO_ID

CDO_OrderDate

ORDERED DRONE COMPONENT

*CDO_ID
*DCM_ID

ODC_QuantOrdered

DRONE BASE MODEL

*DBM_ID

DBM_ModelName
DBM_ModelClass
DBM_Length
DBM_Wingspan
DBM_Weight
DBM_TopSpeed
DBM_MaxDuration
DBM_MaxCeiling
DBM_MotorType
DBM_BatteryType
DBM_CameraType
DBM_Price

DRONE COMPONENT

*DCM_ID

DCM_Name
DCM_Description
DCM_Type
DCM_Weight
DCM_Price

Using the ERD to Show Business Rules (1 of 3)

- Business rules are constraints that are followed when the system is in operation.
- ERD symbols can show when one instance of an entity must exist for an instance of another to exist
  - The customer instance must exist before custom drone order instances can be created for that customer
  - The drone component instance must exist before ordered drone components instances can be created for that drone component

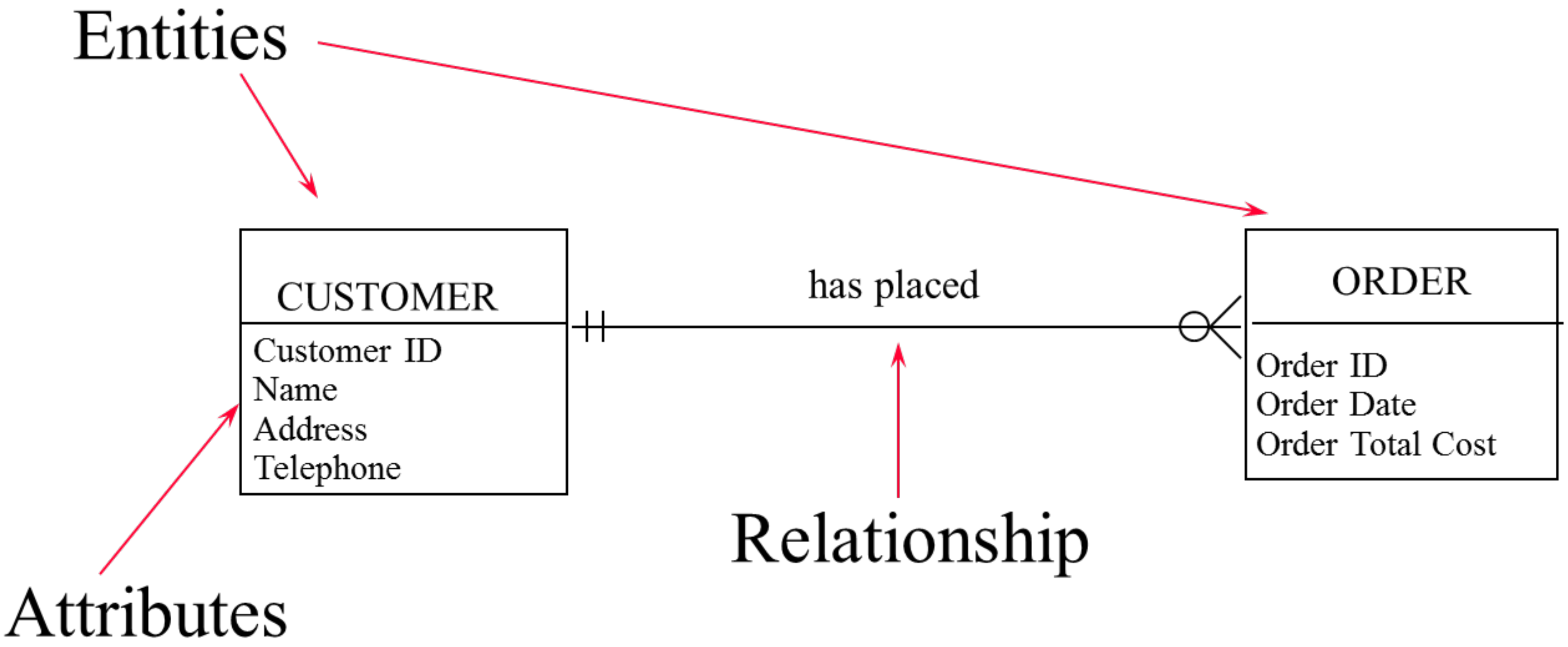
Using the ERD to Show Business Rules (2 of 3)

- ERD symbols can show when one instance of an entity can be related to only one or to many instances of another entity
  - One customer instance can create many custom drone order instances; each custom drone order instance is created by only one customer instance
  - A drone can be included on many chemical requests; a chemical request is for only one chemical

Using the ERD to Show Business Rules (3 of 3)

- ERD symbols show when the existence of an entity instance is optional for a related entity instance
    - A customer instance may exist without being included on any custom drone order instances

An ERD Example

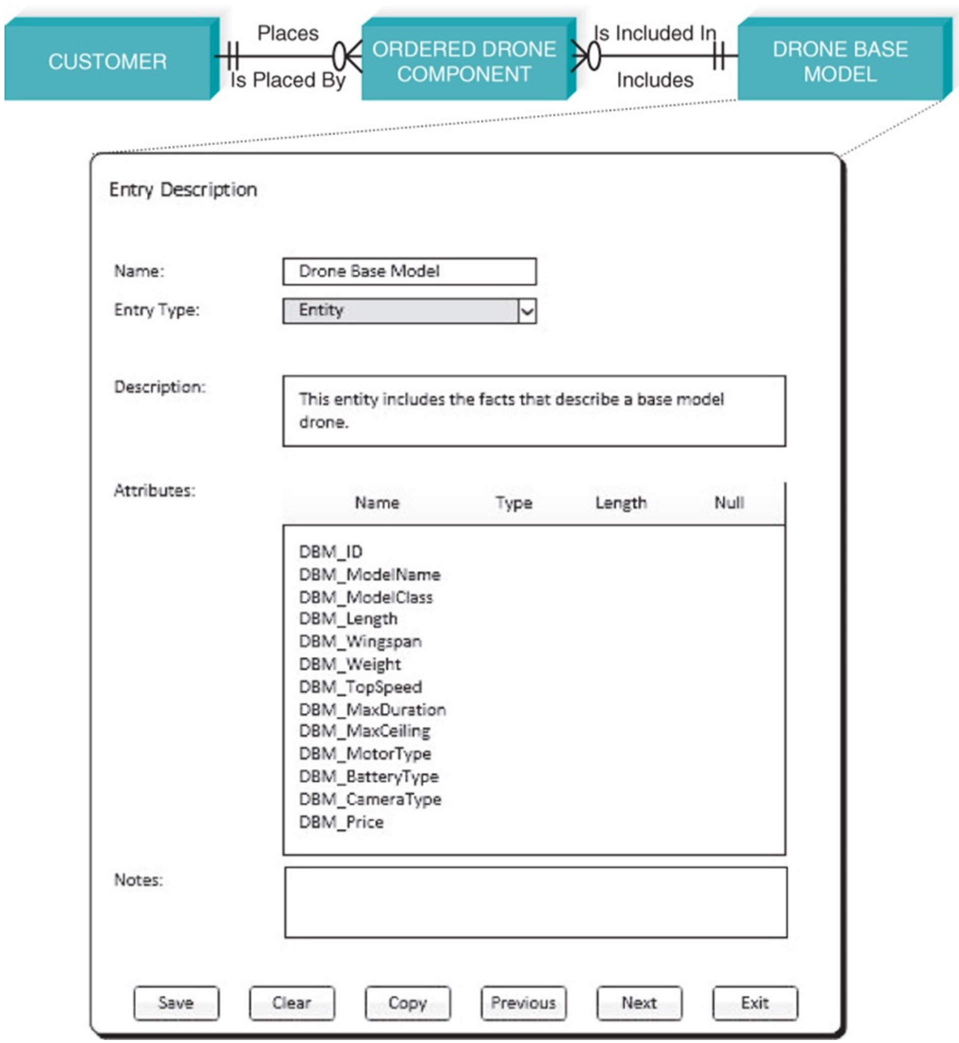


Entity

- A person, place, event, or thing about which data is collected
- Must be multiple occurrences to be an entity

Person	Place	Object	Event	Concept
STUDENT	DORM	BOOK	LIBRARY CHECK OUT	COURSE

CASE Entry for Entity



Entry Description

Name: Drone Base Model

Entry Type: Entity

Description: This entity includes the facts that describe a base model drone.

Attributes:

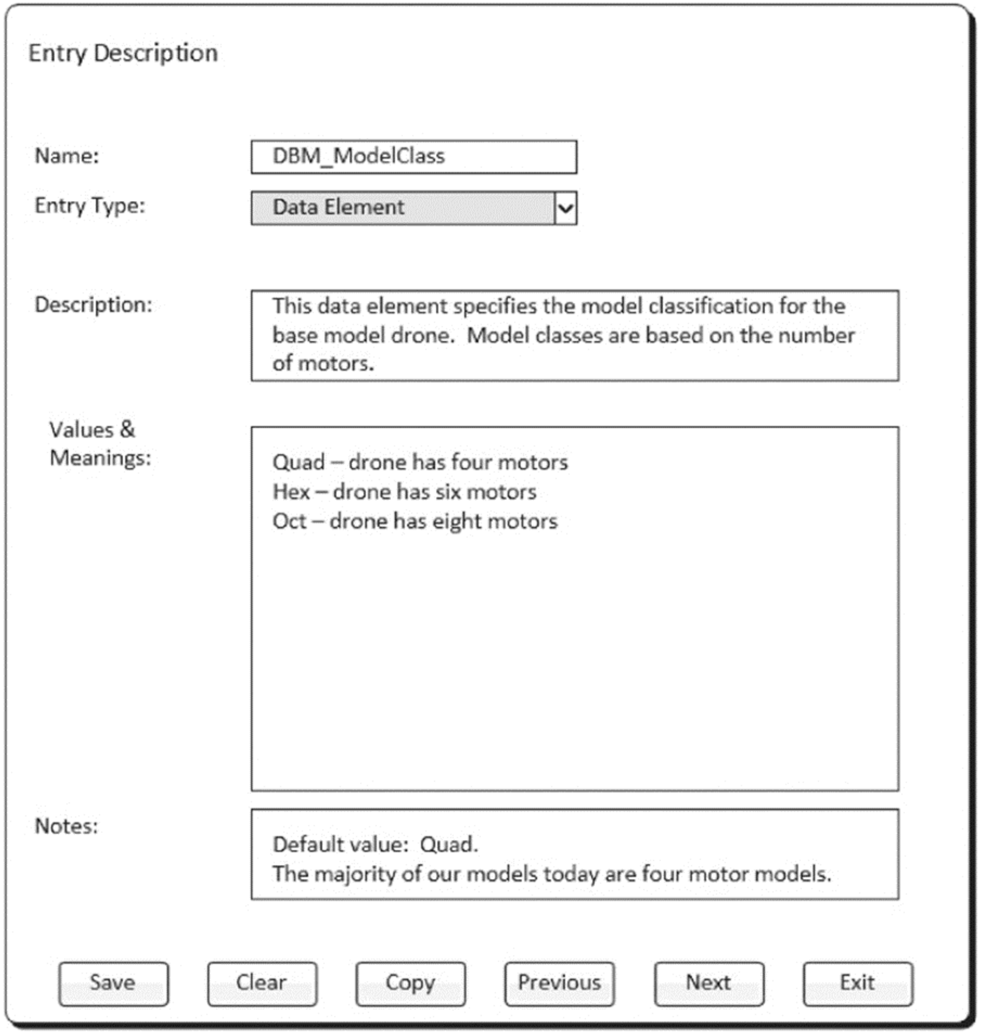
Name	Type	Length	Null
DBM_ID			
DBM_ModelName			
DBM_ModelClass			
DBM_Length			
DBM_Wingspan			
DBM_Weight			
DBM_TopSpeed			
DBM_MaxDuration			
DBM_MaxCeiling			
DBM_MotorType			
DBM_BatteryType			
DBM_CameraType			
DBM_Price			

Notes:

Attributes

- Information captured about an entity
- Only those used by the organization should be included in the model
- Attribute names are nouns
- Sometimes entity name is added at the beginning of the attribute name for clarity

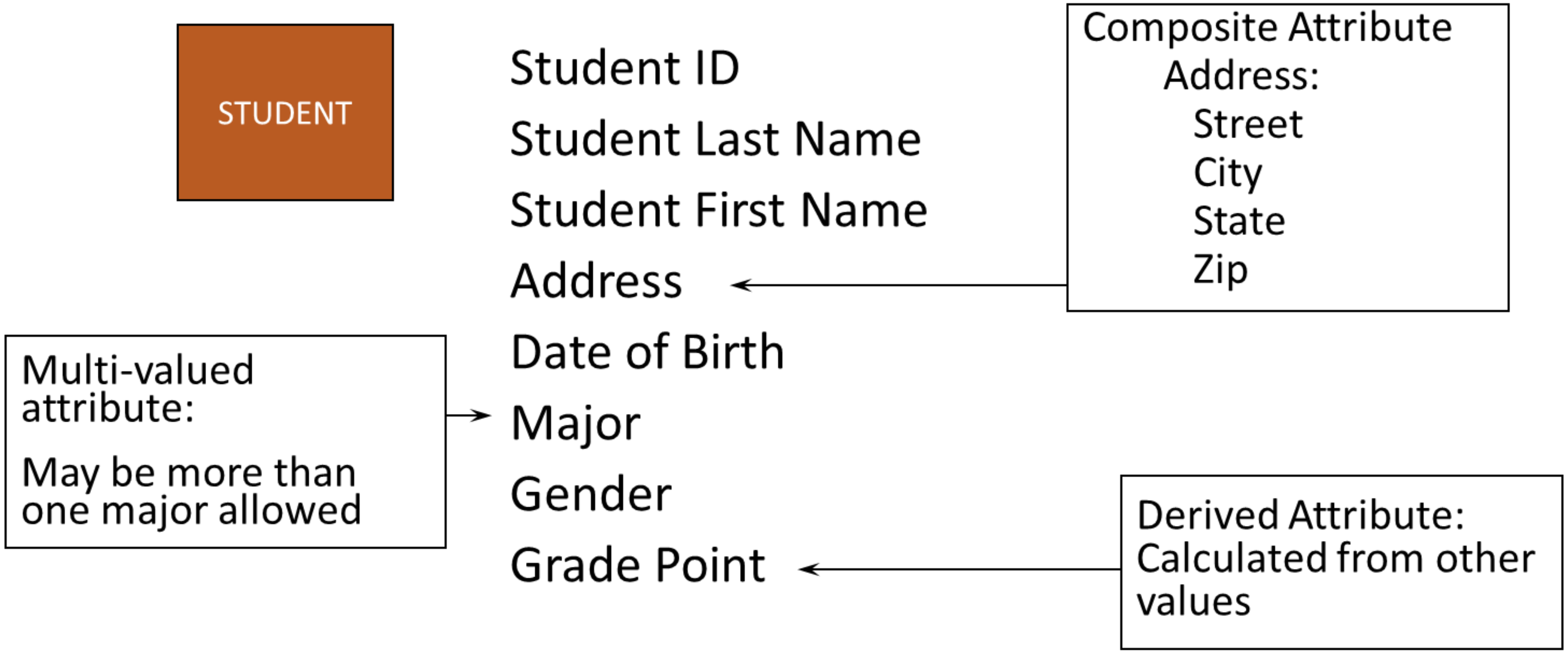
CASE Entry for Attribute



Entry Description

Name:	DBM_ModelClass
Entry Type:	Data Element
Description:	This data element specifies the model classification for the base model drone. Model classes are based on the number of motors.
Values & Meanings:	Quad – drone has four motors Hex – drone has six motors Oct – drone has eight motors
Notes:	Default value: Quad. The majority of our models today are four motor models.

Attributes – Understand Them!



STUDENT

Student ID

Student Last Name

Student First Name

Address

Date of Birth

Major

Gender

Grade Point

Composite Attribute
Address:
Street
City
State
Zip

Multi-valued attribute:

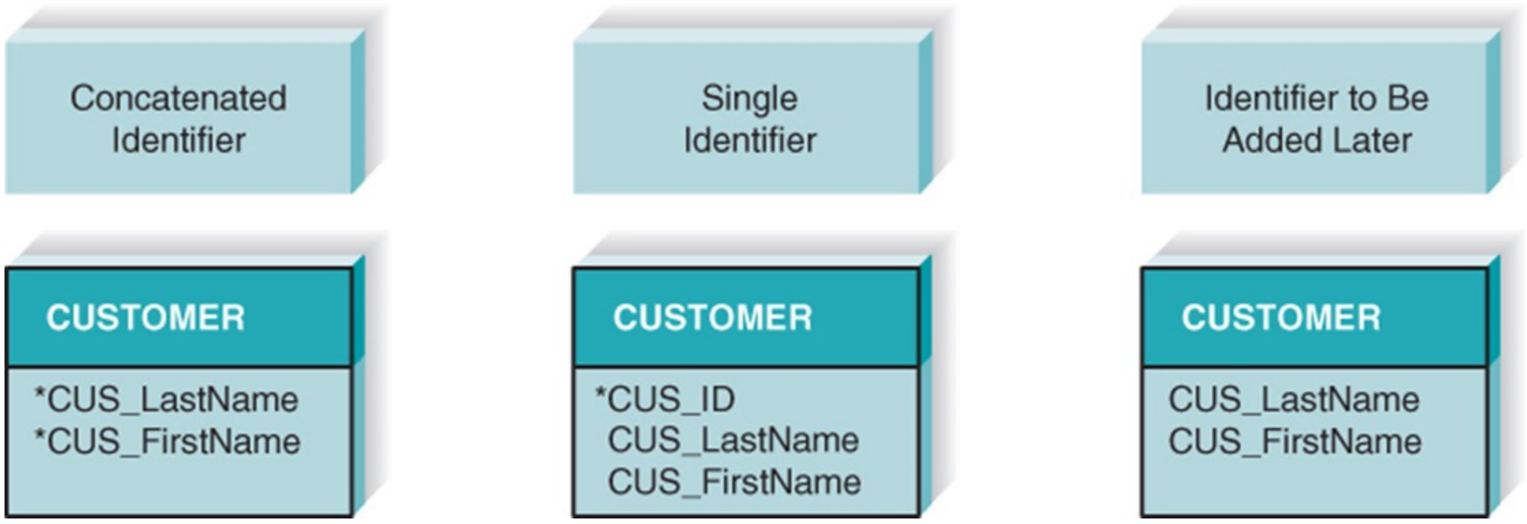
May be more than one major allowed

Derived Attribute: Calculated from other values

Identifier Types (1 of 2)

- One or more attributes can serve as the entity *identifier*, uniquely identifying each entity instance
- Concatenated identifier consists of several attributes
- An identifier may be ‘artificial,’ such as creating an ID number
- Final decision on identifiers may postponed to the Design Phase

Identifier Types (2 of 2)



Concatenated Identifier

CUSTOMER

*CUS_LastName
*CUS_FirstName

Single Identifier

CUSTOMER

*CUS_ID
CUS_LastName
CUS_FirstName

Identifier to Be Added Later

CUSTOMER

CUS_LastName
CUS_FirstName

Relationships

- Associations between entities
- The first entity in the relationship is the *parent* entity; the second entity in the relationship is the *child* entity
- Relationships should have active verb names
- Relationships go in both directions

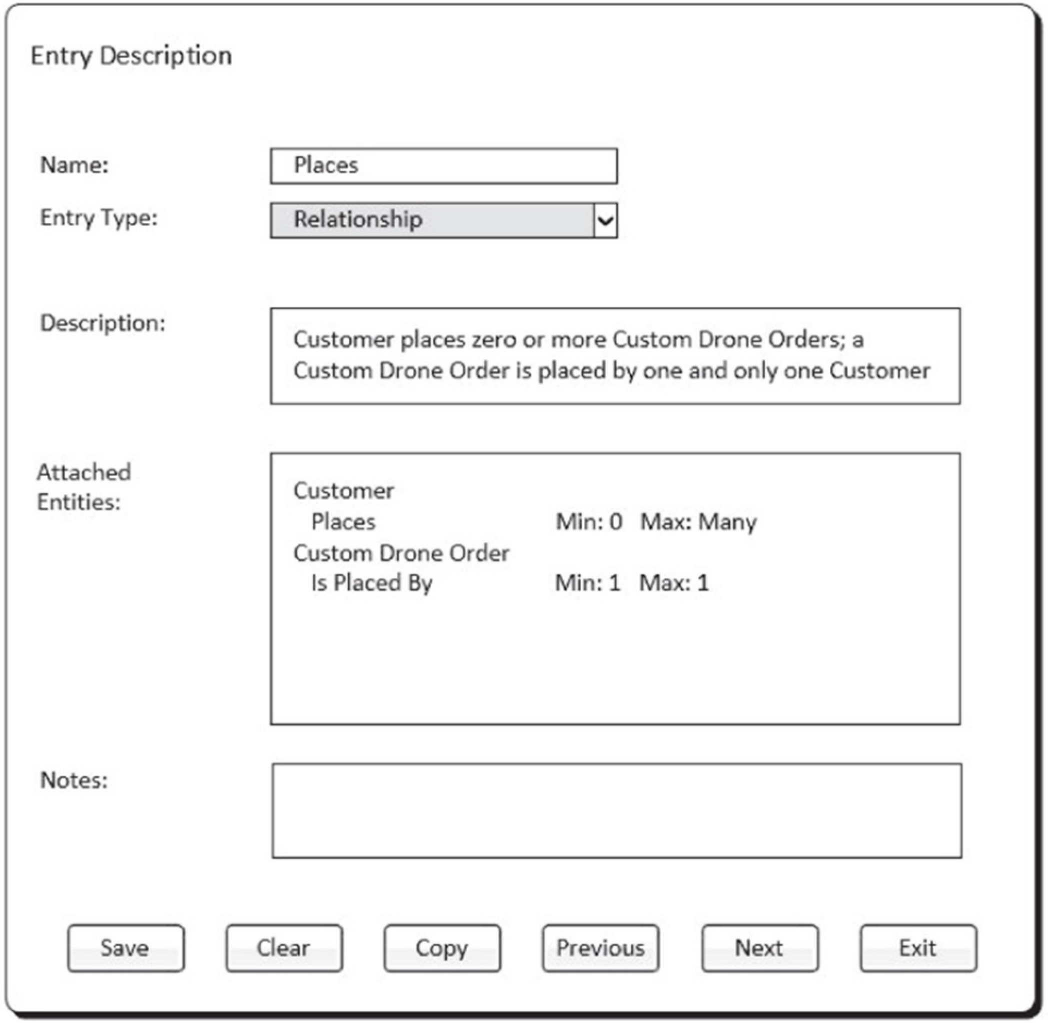
Cardinality

- Refers to the number of times instances in one entity can be related to instances in another entity
  - One instance in an entity refers to one and only one instance in the related entity (1:1)
  - One instance in an entity refers to one or more instances in the related entity (1:N)
  - One or more instances in an entity refer to one or more instances in the related entity (M:N)

Modality

- Refers to whether or not an instance of a child entity can exist without a related instance in the parent entity
  - **Not Null** - an instance in the related entity must exist for an instance in another entity to be valid
  - **Null** - no instance in the related entity is necessary for an instance in another entity to be valid

CASE Entry for Relationship



Entry Description

Name: Places

Entry Type: Relationship

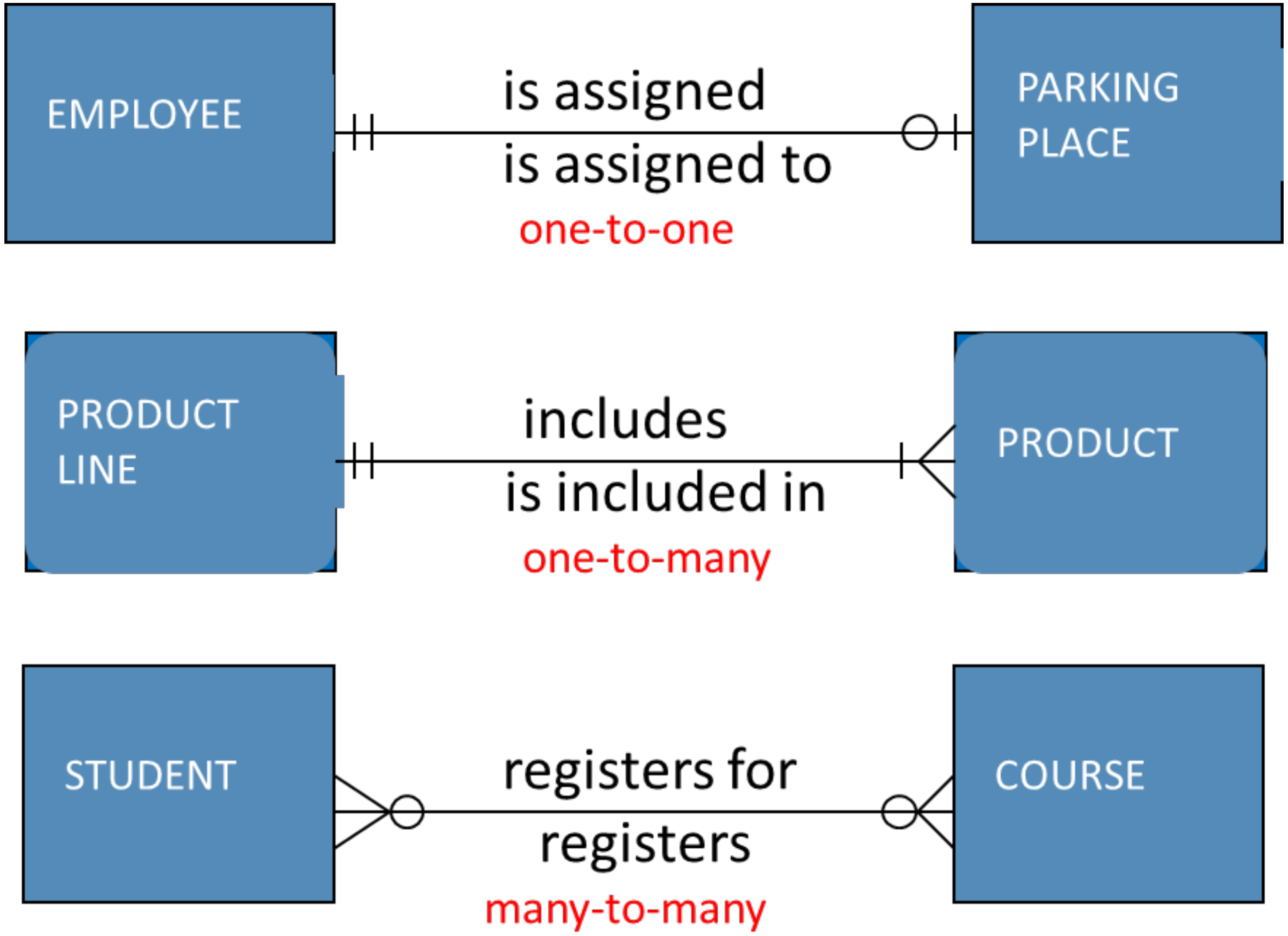
Description: Customer places zero or more Custom Drone Orders; a Custom Drone Order is placed by one and only one Customer

Attached Entities:

Customer		
Places	Min: 0	Max: Many
Custom Drone Order		
Is Placed By	Min: 1	Max: 1

Notes:

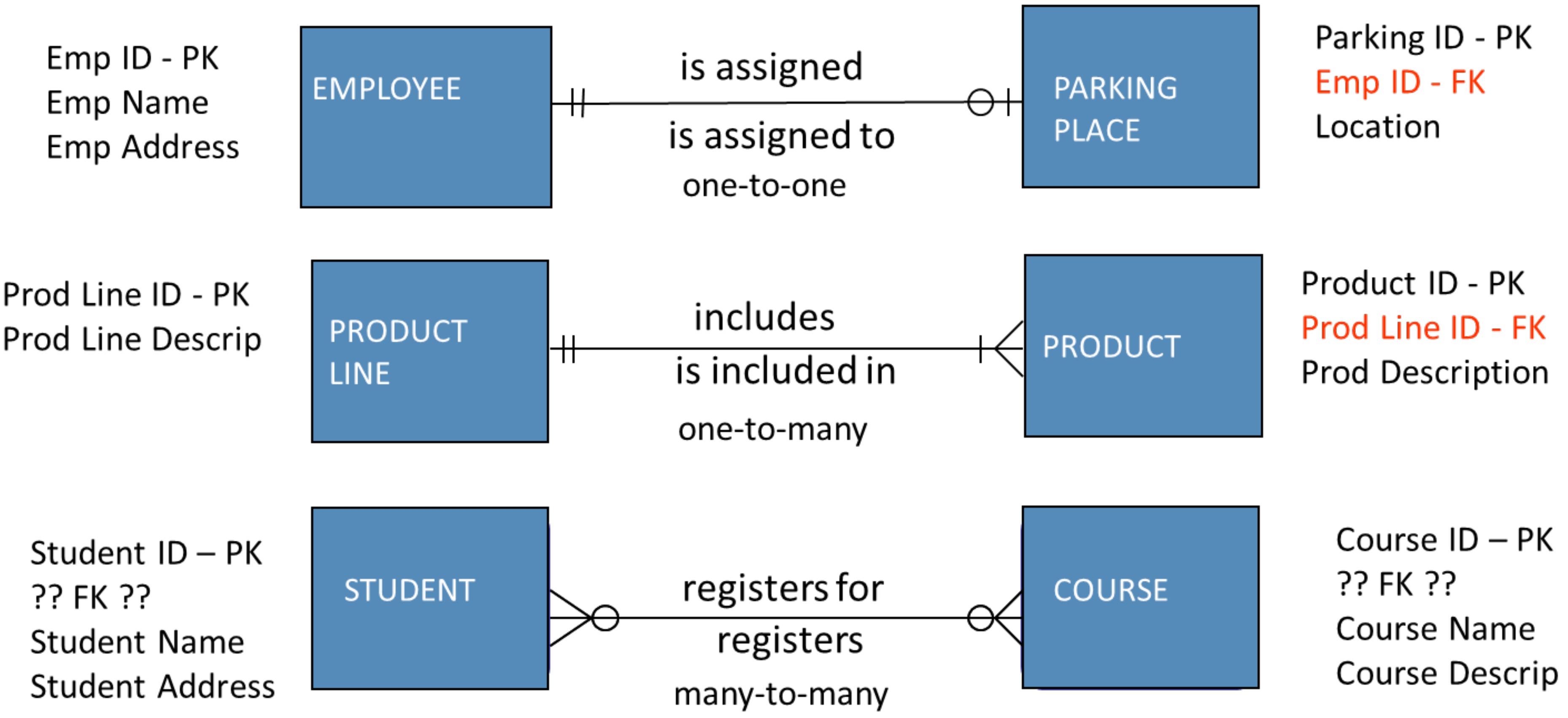
Binary Relationships



Foreign Keys (1 of 2)

- A relationship implies that instances of one entity are related to instances of another entity
- The primary key of one entity is *migrated* into the other entity as a *foreign* key.
- A foreign key is a primary key of one entity that is contributed to (duplicated in) another entity for the purpose of identifying instances of a relationship. A foreign key (always in a child entity) always matches the primary key (in a parent entity).

Foreign Keys (2 of 2)



Emp ID - PK
Emp Name
Emp Address

EMPLOYEE

is assigned

is assigned to

one-to-one

PARKING PLACE

Parking ID - PK
Emp ID - FK
Location

Prod Line ID - PK
Prod Line Descrip

PRODUCT LINE

includes

is included in

one-to-many

PRODUCT

Product ID - PK
Prod Line ID - FK
Prod Description

Student ID – PK
?? FK ??
Student Name
Student Address

STUDENT

registers for

registers

many-to-many

COURSE

Course ID – PK
?? FK ??
Course Name
Course Descrip

Creating an ERD

HOW ERDS ARE DEVELOPED

Overview

- Drawing the ERD is an iterative process of trial and revision
- ERDs can become quite complex
- Steps in building ERDs…
  - Identify the entities
  - Add appropriate attributes for each entity
  - Draw the relationships that connect associated entities

Identify the Entities

- Identify major categories of information
  - If available, check the process models for data stores, external entities, and data flows
  - Check the major inputs and outputs from the use cases
- Verify that there is more than one instance of the entity that occurs in the system

Add Appropriate Attributes

- Identify attributes of the entity that are relevant to the system under development
  - Check the process model repository entries for details on data flows and data stores
  - Check the data requirements of the requirements definition
  - Interview knowledgeable users
  - Perform document analysis on existing forms and reports
- Select the entity's candidate identifier (final decision may be postponed until Design phase)

Draw the Relationships

- Start with an entity and identify all entities with which it shares relationships
- Describe the relationship with the appropriate verb phrase
- Determine the cardinality and modality by discussing the business rules with knowledgeable users

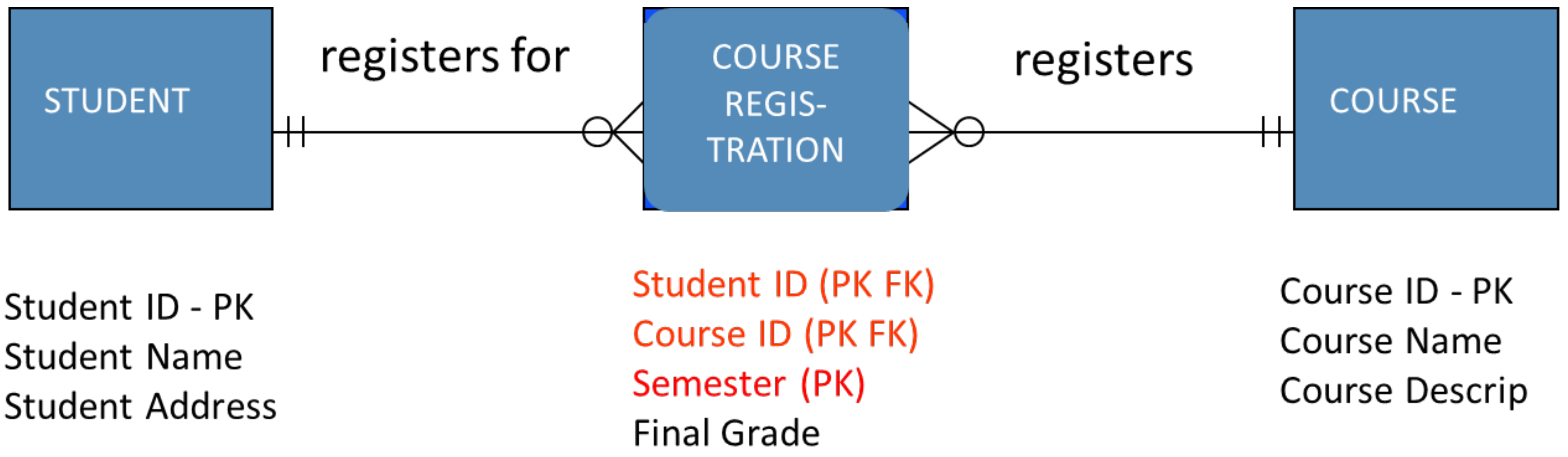
ERD Building Tips

- Data stores of the DFD generally correspond to entities
- Only include entities with more than one instance
- Don’t include entities associated with implementation of the system (e.g., archive files of older data). They will be added later.

Advanced Syntax - Intersection Entities

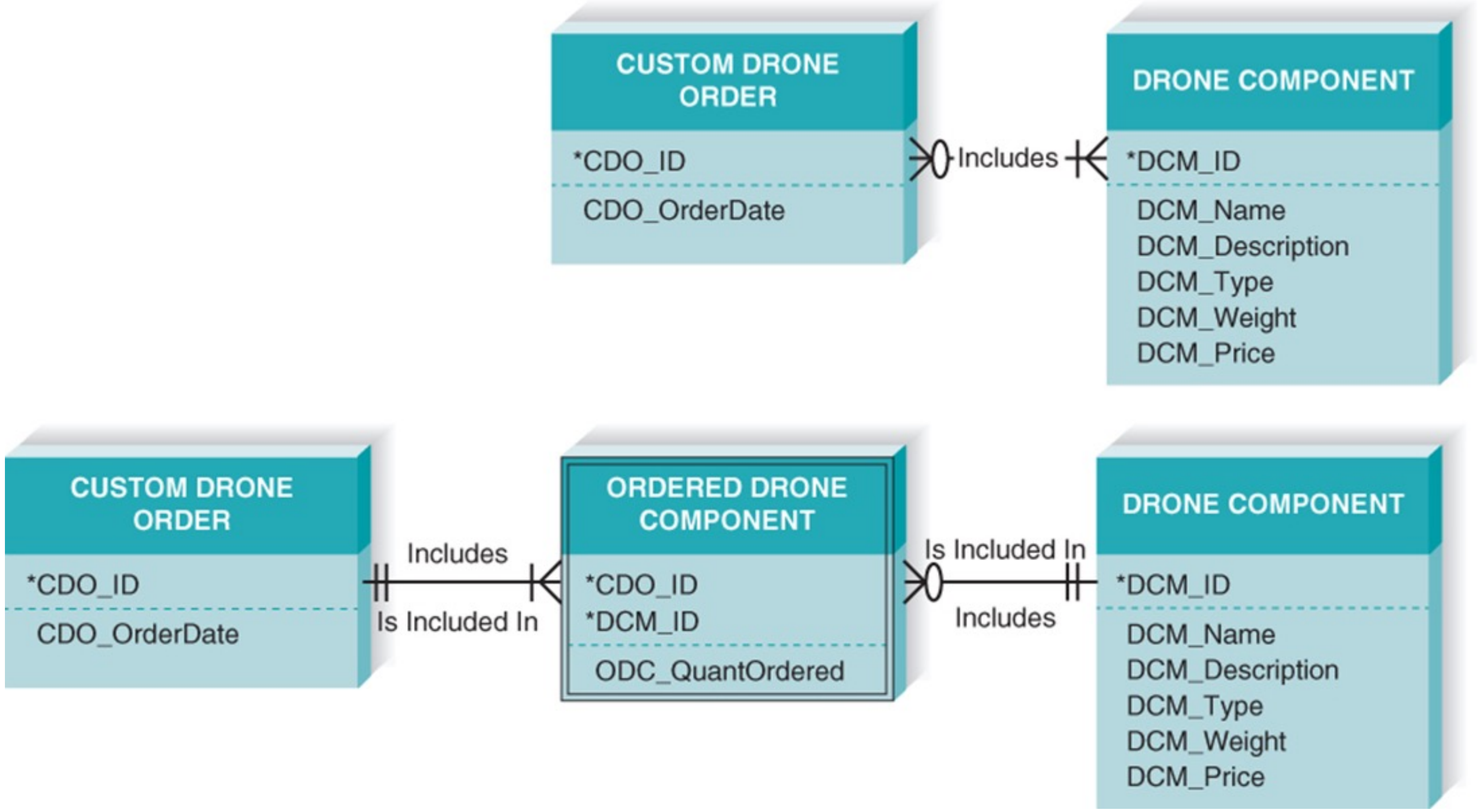
- A new entity is created to store information about two entities sharing an M:N relationship
  - Remove the M:N relationship between two entities and insert new entity between them
  - Create two 1:N relationships: original entities are parents to the new child intersection entity
  - Name the intersection entity
  - Migrate parent entity primary keys to new entity as foreign keys (possibly also concatenated primary key)

Resolving M:N with an Intersection Entity



Resolving M:N Relationship

Additional Example



Validating an ERD

ENSURING A QUALITY DATA MODEL

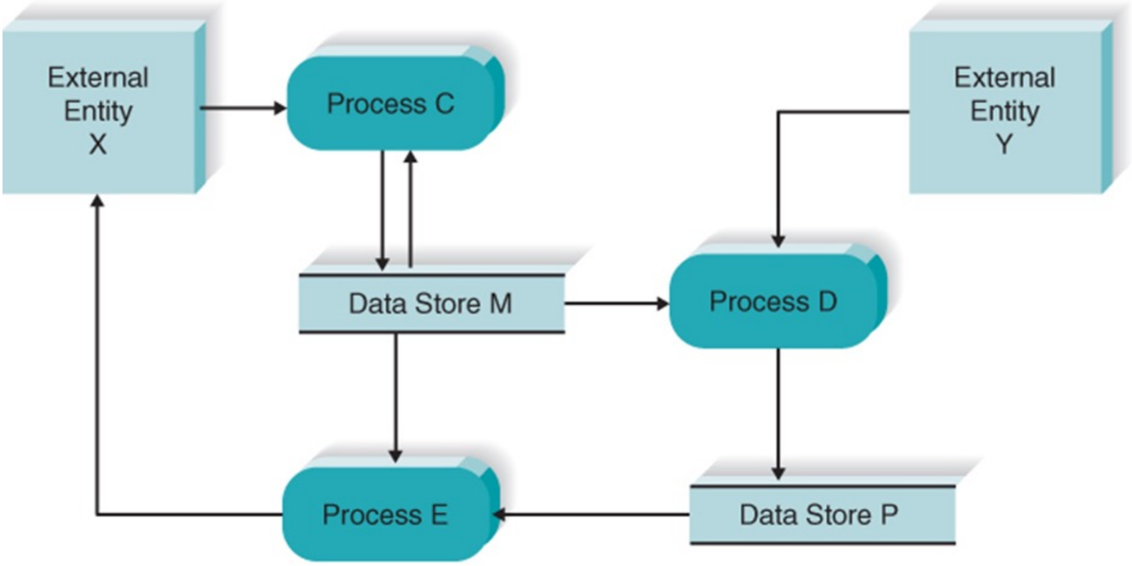
Design Guidelines

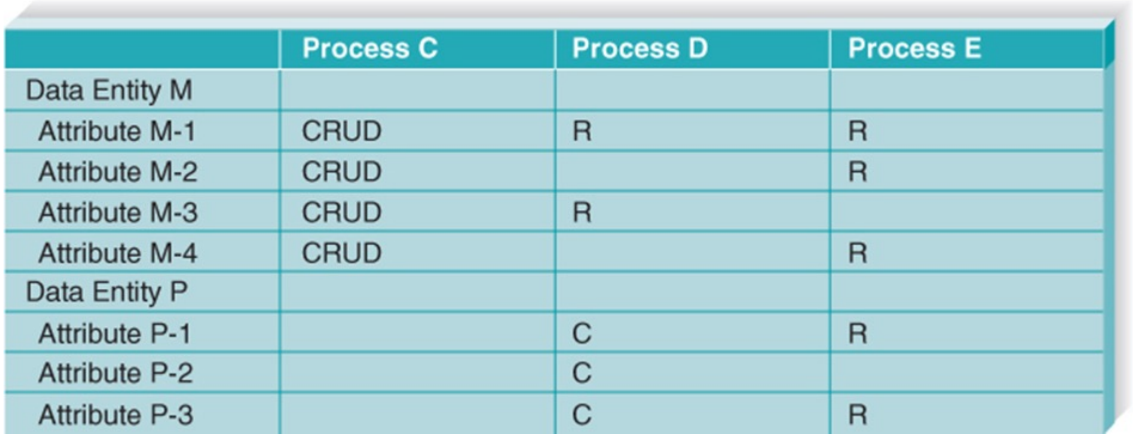
- Best practices rather than rigid rules
- Entities should have many occurrences
- Avoid unnecessary attributes
- Clearly label all components
- Apply correct cardinality and modality
- Break attributes into lowest level needed
- Labels should reflect common business terms
- Assumptions should be clearly stated

Balancing ERDs with DFDs

- All analysis activities are interrelated
- Process models contain two data components
  - Data flows and data stores
- The DFD data components need to balance the ERD's data stores (entities) and data elements (attributes)
- Many CASE tools provide features to check for imbalance
- Check that all data stores and elements correspond between models
  - Data that is not used is unnecessary
  - Data that has been omitted results in an incomplete system
- Do not follow thoughtlessly -- check that the models make sense!

Use of a CRUD Matrix





	Process C	Process D	Process E
Data Entity M			
Attribute M-1	CRUD	R	R
Attribute M-2	CRUD		R
Attribute M-3	CRUD	R	
Attribute M-4	CRUD		R
Data Entity P			
Attribute P-1		C	R
Attribute P-2		C	
Attribute P-3		C	R

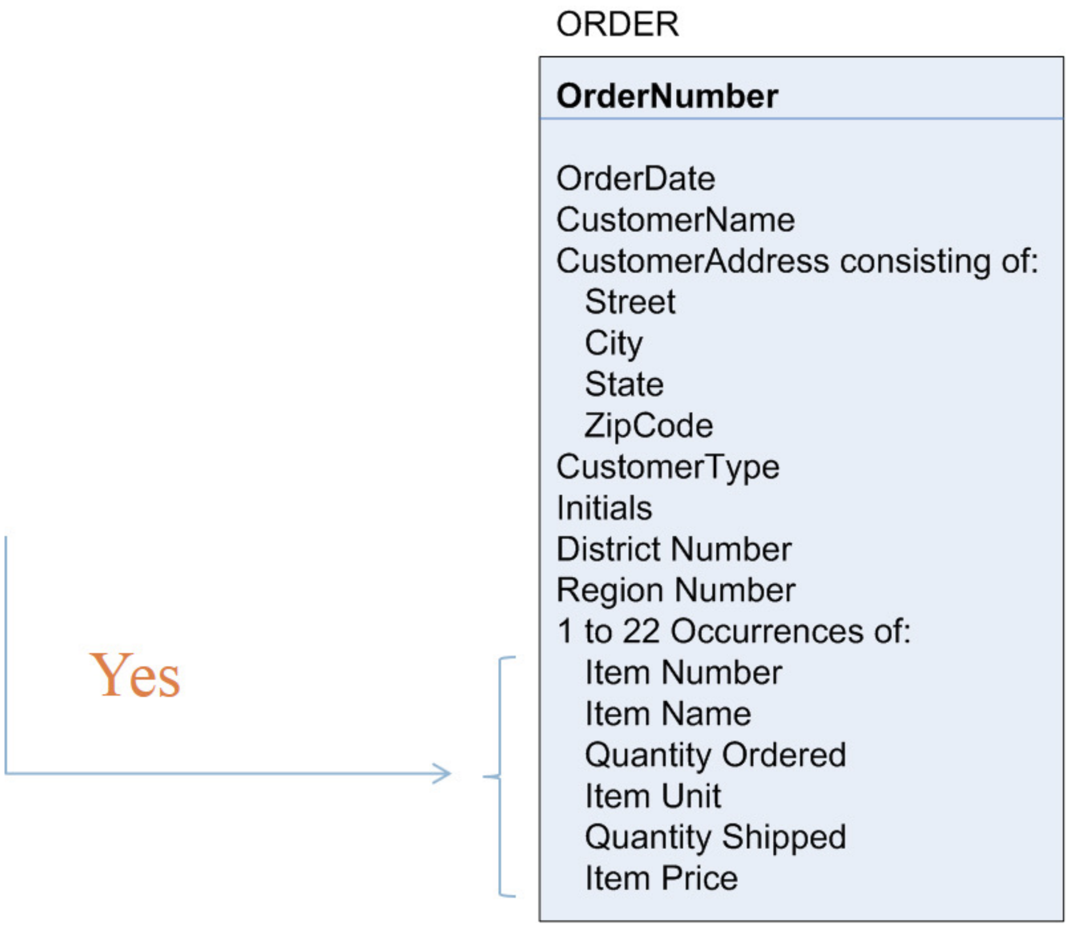
Normalization

- Technique used to validate data models
- Series of rules applied to logical data model to improve its organization
- Three normalization rules are common

Example 1: Unnormalized Entity

Begin with an entity from the logical data model

Do any attributes (or groups of attributes) occur more than once for a single occurrence of the entity?



Yes

ORDER

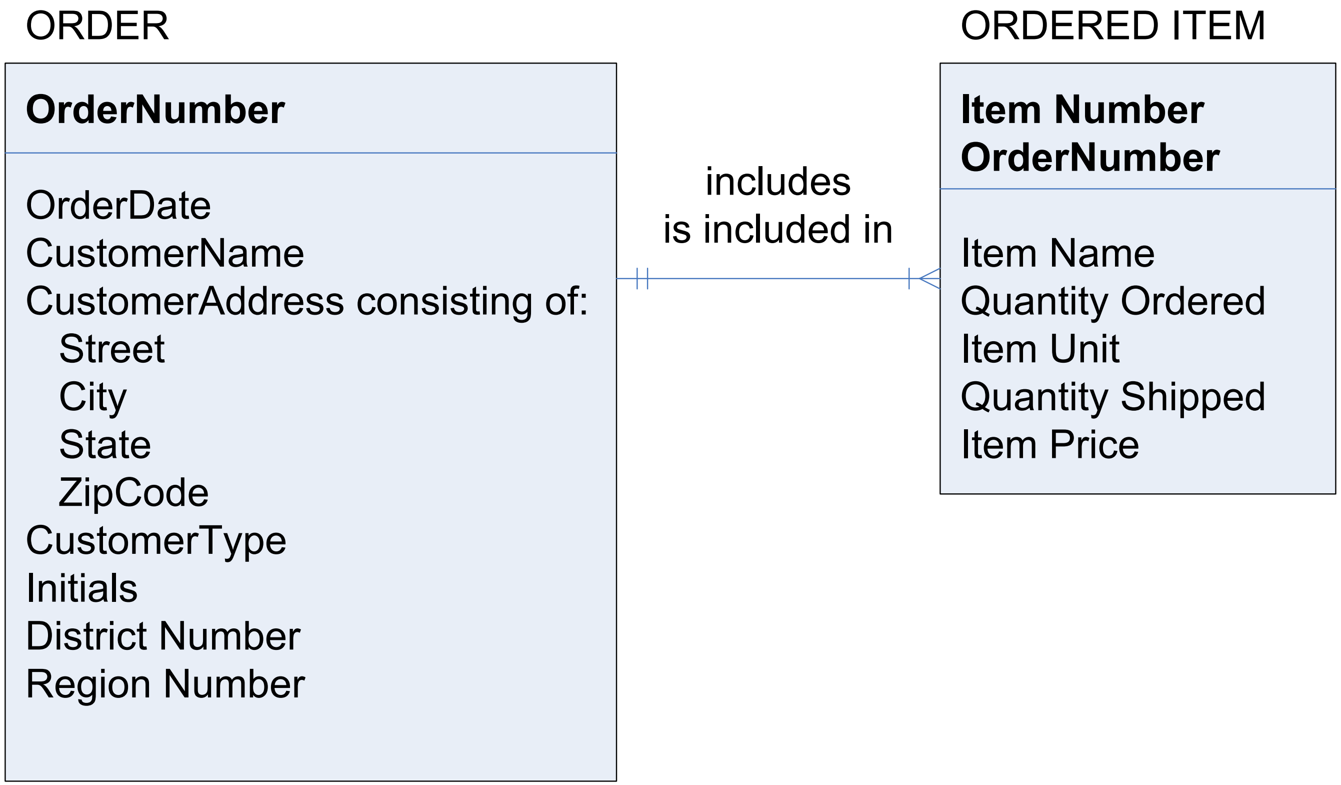
OrderNumber

OrderDate
CustomerName
CustomerAddress consisting of:
  Street
  City
  State
  ZipCode
CustomerType
Initials
District Number
Region Number
1 to 22 Occurrences of:
  Item Number
  Item Name
  Quantity Ordered
  Item Unit
  Quantity Shipped
  Item Price

Example 1: 1st Normal Form

Do any attributes (or groups of attributes) occur more than once for a single occurrence of the entity?

If yes, remove the attributes (or groups) into separate entities.



ORDER

OrderNumber

OrderDate
CustomerName
CustomerAddress consisting of:
Street
City
State
ZipCode
CustomerType
Initials
District Number
Region Number

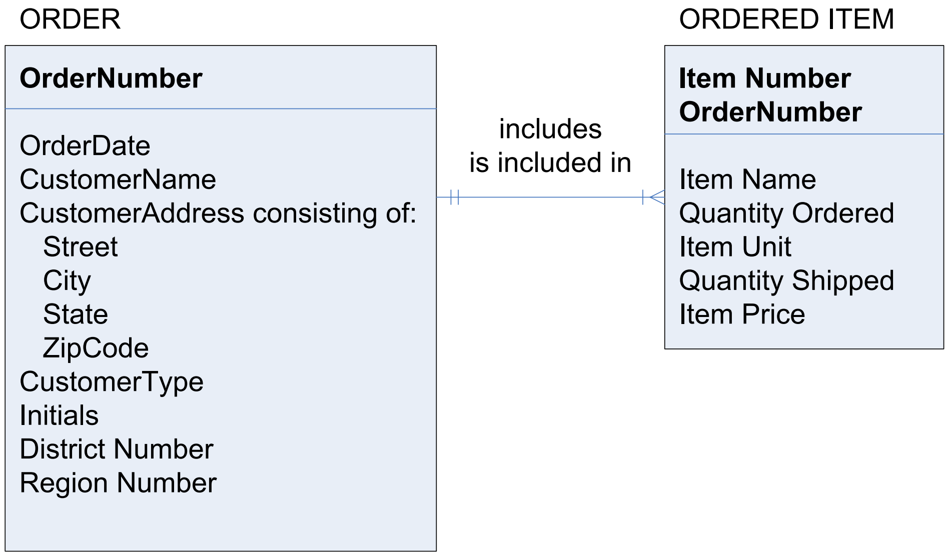
includes
is included in

ORDERED ITEM

Item Number
OrderNumber

Item Name
Quantity Ordered
Item Unit
Quantity Shipped
Item Price

Potential Anomalies with 1st Normal Form



Insert anomaly:

Insert a new Item?

Can’t do without Order Number

Deletion anomaly:

Assume only one order has been made for Item #456. What happens if that order is cancelled?

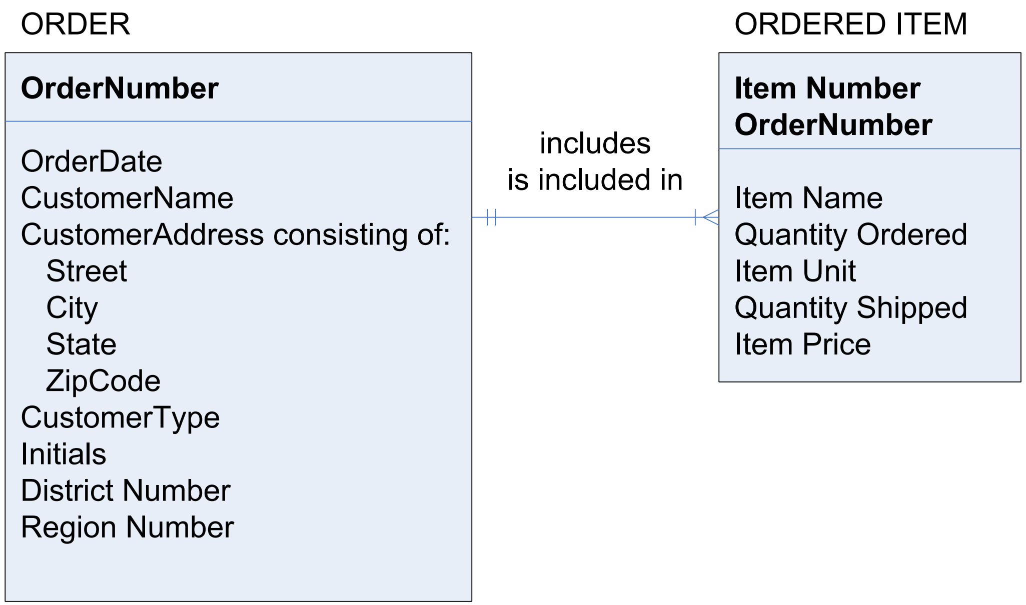
Will lose all information about Item # 456

Update anomaly:

The price of Item # 789 should be changed from $7.99 to $8.95. What problem occurs?

Need to search entire database for all occurrences of Item # 789

Potential Anomalies with 1st Normal Form



ORDER

OrderNumber

OrderDate
CustomerName
CustomerAddress consisting of:
  Street
  City
  State
  ZipCode
CustomerType
Initials
District Number
Region Number

includes
is included in

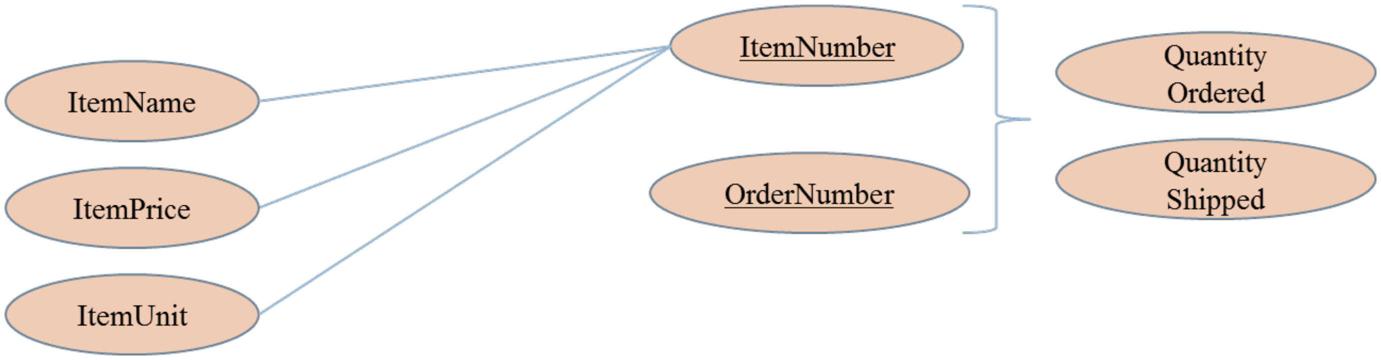
ORDERED ITEM

Item Number
OrderNumber

Item Name
Quantity Ordered
Item Unit
Quantity Shipped
Item Price

Reason anomalies exist?

Several non-key attributes depend only on ItemNumber and NOT on the full primary key (ItemNumber + OrderNumber)

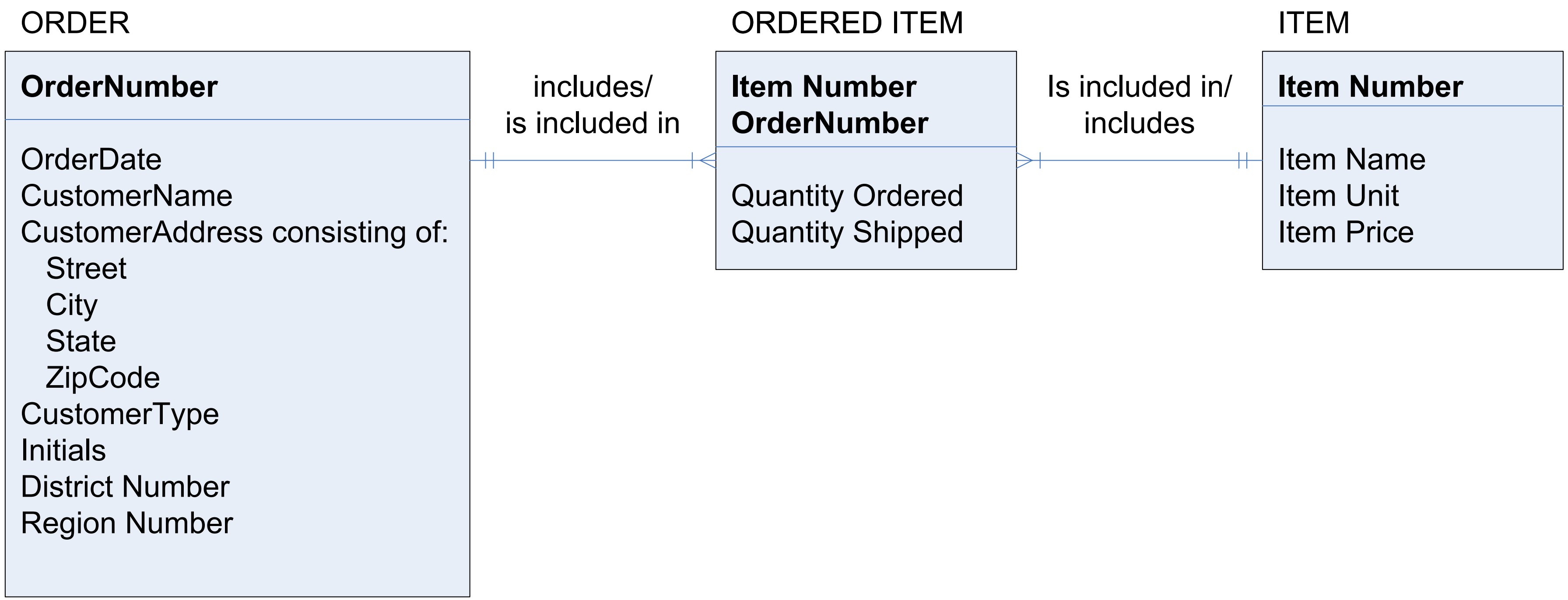


Example 1: 2nd Normal Form

For entities with concatenated keys…

Do any attributes depend on just part of the key rather than the entire key?

If yes, move partially-dependent attributes to a new entity…



ORDER

OrderNumber

OrderDate
CustomerName
CustomerAddress consisting of:
Street
City
State
ZipCode
CustomerType
Initials
District Number
Region Number

includes/
is included in

ORDERED ITEM

Item Number
OrderNumber

Quantity Ordered
Quantity Shipped

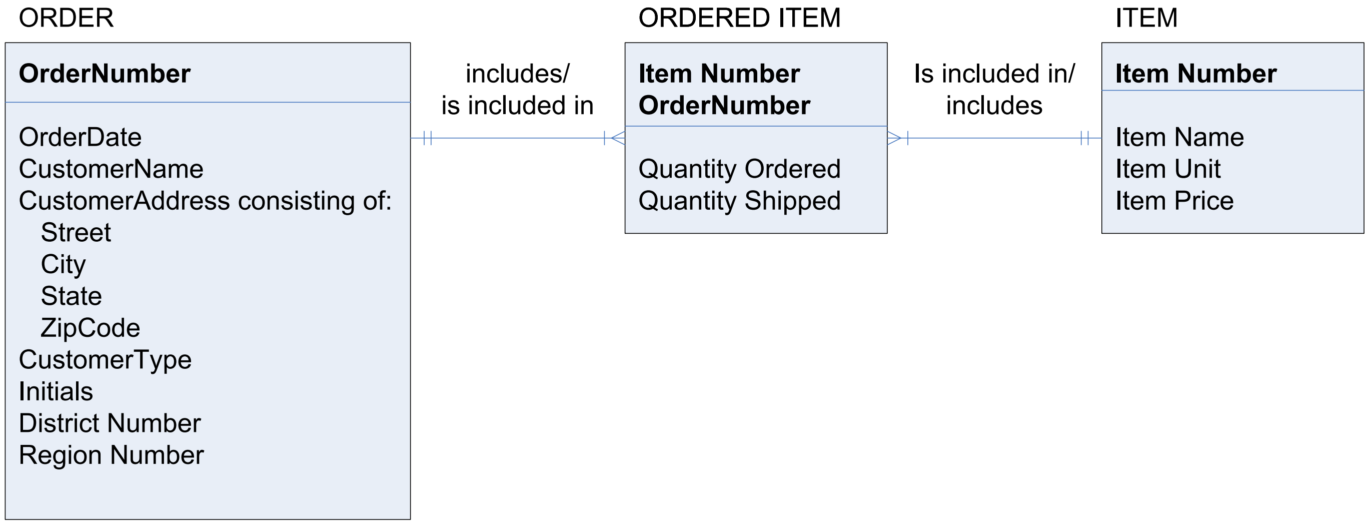
Is included in/
includes

ITEM

Item Number

Item Name
Item Unit
Item Price

Potential Anomalies with 2nd Normal Form



Reason Anomalies Exist:

The ORDER entity contains transitive dependencies

This means that several non-key attributes depend on another non-key attribute, and NOT on the Primary Key

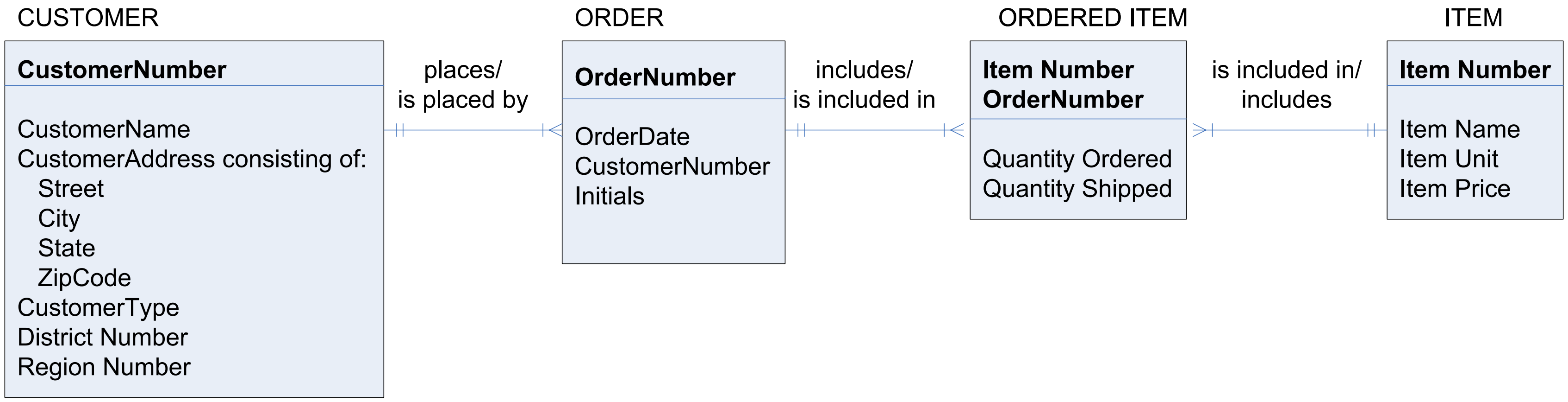
CustomerAddress, CustomerType, DistrictNumber, and RegionNumber depend on the CustomerName, NOT the OrderNumber

Solution: remove these attributes to a separate entity

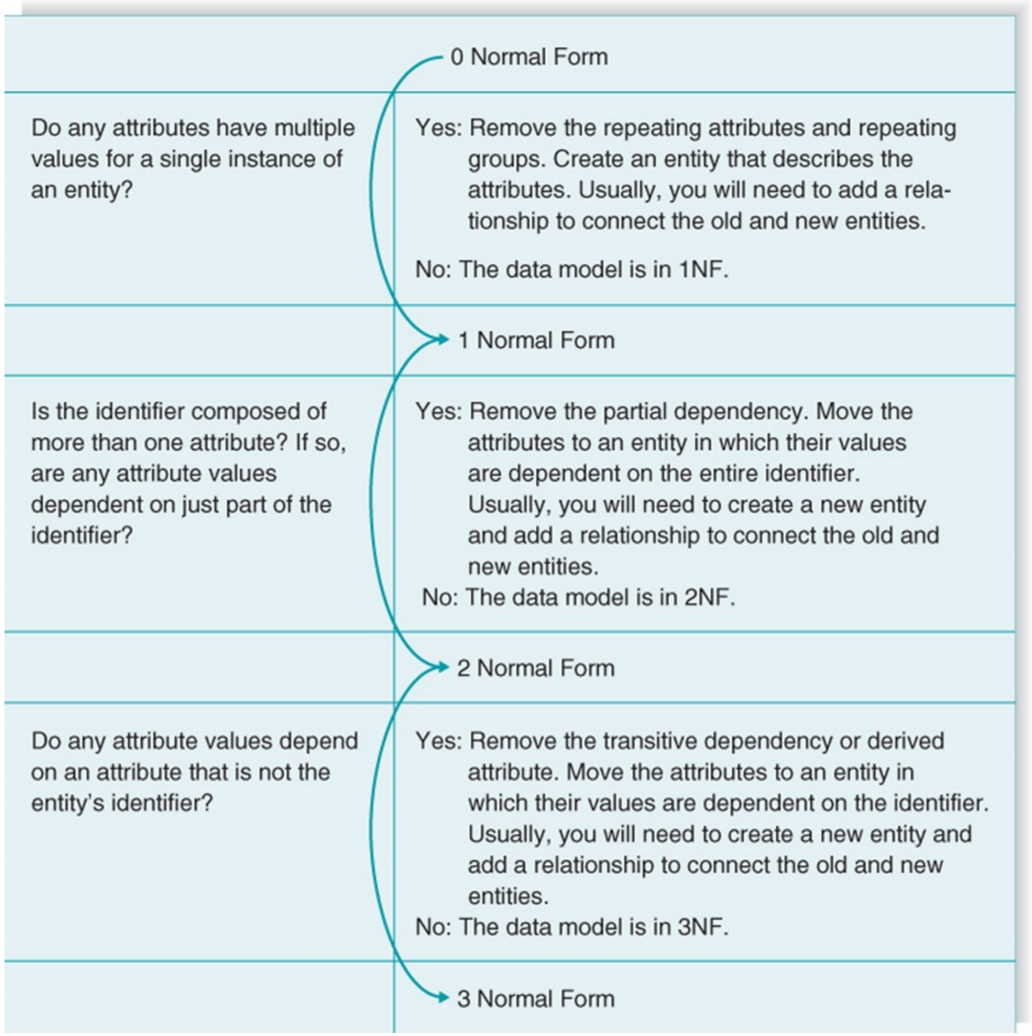
Example 1: 3rd Normal Form

Do any attribute values depend on an attribute that is not the entity’s key?

If yes, move these attributes to a new entity.



Summary of Normalization Steps



	0 Normal Form
Do any attributes have multiple values for a single instance of an entity?	Yes: Remove the repeating attributes and repeating groups. Create an entity that describes the attributes. Usually, you will need to add a relationship to connect the old and new entities. No: The data model is in 1NF.
	1 Normal Form
Is the identifier composed of more than one attribute? If so, are any attribute values dependent on just part of the identifier?	Yes: Remove the partial dependency. Move the attributes to an entity in which their values are dependent on the entire identifier. Usually, you will need to create a new entity and add a relationship to connect the old and new entities. No: The data model is in 2NF.
	2 Normal Form
Do any attribute values depend on an attribute that is not the entity's identifier?	Yes: Remove the transitive dependency or derived attribute. Move the attributes to an entity in which their values are dependent on the identifier. Usually, you will need to create a new entity and add a relationship to connect the old and new entities. No: The data model is in 3NF.
	3 Normal Form

After reading and studying this chapter, you should be able to: (1 of 2)

- Define the meaning and purpose of the entity and relationship shown on an entity relationship diagram (ERD).
- Explain the meaning and purpose of attributes included in a data model.
- Explain what is meant by an entity’s identifier.
- Explain the meaning of the cardinality and modality of a relationship.

After reading and studying this chapter, you should be able to: (2 of 2)

- Explain the concept of metadata and how it is compiled in the project repository.
- Discuss the process used to create a data model.
- Describe how to ensure that the process model and data model are balanced through the use of the CRUD matrix.
- Discuss how the normalization process is performed and how it contributes to the quality of the data model (from chapter appendix).

Copyright