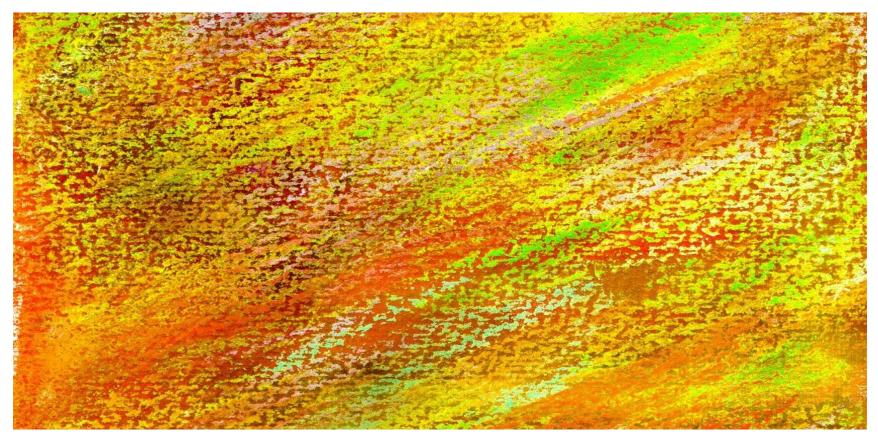


# DAVID M. KROENKE and DAVID J. AUER DATABASE CONCEPTS, 6<sup>th</sup> Edition

# Chapter Four Data Modeling and the Entity-Relationship Model





### Chapter Objectives

- Learn the basic stages of database development
- Understand the purpose and role of a data model
- Know the principal components of the E-R data model
- Understand how to interpret traditional E-R diagrams
- Understand how to interpret the Information Engineering (IE) model's Crow's Foot E-R diagrams
- Learn to construct E-R diagrams
- Know how to represent 1:1, 1:N, N:M, and binary relationships with the E-R model



# Chapter Objectives (Cont'd)

- Understand two types of weak entities and know how to use them
- Understand nonidentifying and identifying relationships and know how to use them
- Know how to represent subtype entities with the E-R model
- Know how to represent recursive relationships with the E-R model
- Learn how to create an E-R diagram from source documents



# Three Stages of Database Development

- The three stages of database development are:
  - Requirements Analysis Stage
  - Component Design Stage
  - Implementation Stage
- These three stages are part of the five stage Systems Development Life Cycle (SDLC) model—See online Appendix F— Getting Started in Systems Analysis and Design, for more information



### The Requirements Analysis Stage

- Sources of requirements
  - User Interviews
  - Forms
  - Reports
  - Queries
  - Use Cases
  - Business Rules



# Requirements Become the E-R Data Model

- After the requirements have been gathered, they are transformed into an Entity Relationship (E-R) Data Model.
- The most important elements of E-R Models are:
  - Entities
  - Attributes
  - Identifiers
  - Relationships



### Entity Class versus Entity Instance

- An entity class is a description of the structure and format of the occurrences of the entity.
- An entity instance is a specific occurrence of an entity within an entity class.



#### **Entity Class and Entity Instance**

#### ITEM

ItemNumber
Description
Cost
ListPrice
QuantityOnHand

**Entity Class** 

1100 100 amp panel \$127.50 \$170.00 14 2000 Door handle set \$52.50 \$39.38 0

Two Entity Instances

Figure 4-2: The ITEM Entity and Two Entity Instances



#### **Attributes**

- Entities have attributes that describe the entity's characteristics:
  - ProjectName
  - StartDate
  - ProjectType
  - ProjectDescription
- Attributes have a data type and properties.



#### Identifiers

- Entity instances have identifiers.
- An identifier will identify a particular instance in the entity class:
  - SocialSecurityNumber
  - StudentID
  - EmployeeID



# Identifier Types

- Uniqueness
  - Identifiers may be unique or nonunique.
  - If the identifier is unique, the data value for the identifier must be unique for all instances.
- Composite
  - A composite identifier consists of two or more attributes.
    - E.g., OrderNumber & LineItemNumber are both required.



#### Levels of Entity Attribute Display

#### ITEM

**ItemNumber** 

Description
Cost
ListPrice
QuantityOnHand

(a) Entity with All Attributes

#### ITEM

**ItemNumber** 

(b) Entity with Identifier Attribute Only

#### **ITEM**

(c) Entity with No Attributes

Figure 4-3: Levels of Entity Attribute Display

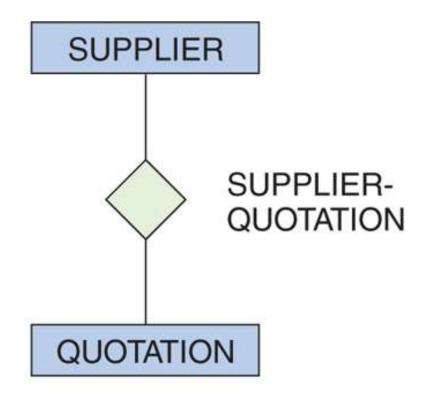


## Relationships

- Entities can be associated with one another in relationships.
- Relationship degree defines the number of entity classes participating in the relationship:
  - Degree 2 is a binary relationship.
  - Degree 3 is a ternary relationship.



### Degree 2 Relationship: Binary

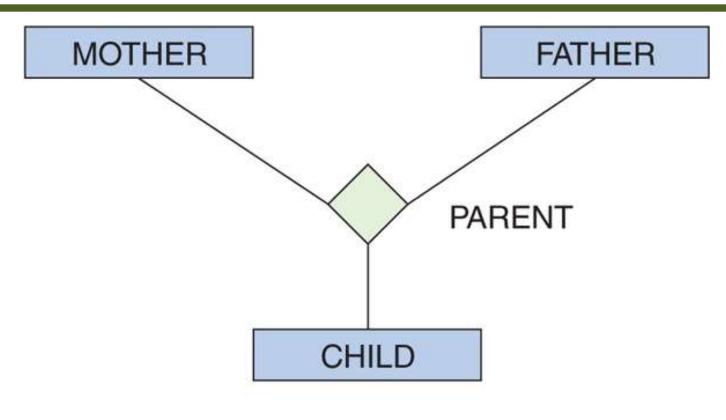


(a) Binary Relationship

Figure 4-4: Example Relationships



### Degree 3 Relationship: Ternary



(b) Ternary Relationship

Figure 4-4: Example Relationships



### One-to-One Binary Relationship

#### • 1:1 (one-to-one)

- A single entity instance in one entity class is related to a single entity instance in another entity class.
  - An employee may have no more than one locker; and
  - A locker may only be accessible by one employee LOCKER-ASSIGNMENT



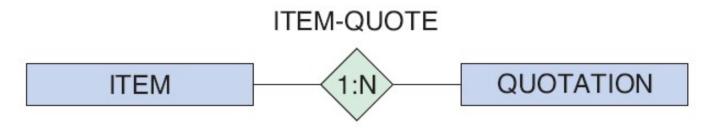
(a) One-to-One Relationship
Figure 4-5: Three Types of Binary Relationships



### One-to-Many Binary Relationship

#### 1:N (one-to-many)

- A single entity instance in one entity class is related to many entity instances in another entity class.
  - A quotation is associated with only one item; and
  - An item may have several quotations



(b) One-to-Many Relationship

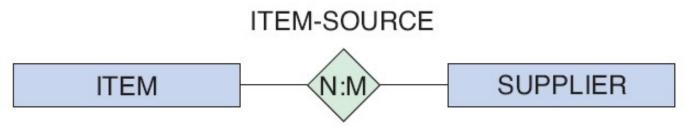
Figure 4-4: Three Types of Binary Relationships



### Many-to-Many Binary Relationship

#### N:M (many-to-many)

- Many entity instances in one entity class is related to many entity instances in another entity class:
  - a supplier may supply several items; and
  - a particular item may be supplied by several suppliers.



(c) Many-to-Many Relationship
Figure 4-5: Three Types of Binary Relationships



## Maximum Cardinality

- Relationships are named and classified by their cardinality, which is a word that means count.
- Each of the three types of binary relationships shown above have different maximum cardinalities.
- Maximum cardinality is the maximum number of entity instances that may participate in a relationship instance one, many, or some other fixed number.



# Minimum Cardinality

- Minimum cardinality is the minimum number of entity instances that must participate in a relationship instance.
- These values typically assume a value of zero (optional) or one (mandatory).



# Cardinality Example

- Maximum cardinality is many for both ITEM and SUPPLIER.
- Minimum cardinality is zero (optional) for ITEM and one (mandatory) SUPPLIER.
  - A SUPPLIER does not have to supply an ITEM.
  - An ITEM must have a SUPPLIER.

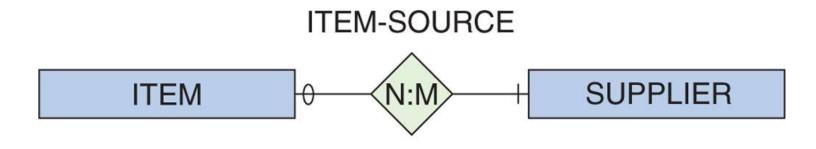


Figure 4-6: A Relationship with Minimum Cardinalities



### Entity-Relationship Diagrams

- The diagrams in previous slides are called entity-relationship diagrams.
  - Entity classes are shown by rectangles.
  - Relationships are shown by diamonds.
  - The maximum cardinality of the relationship is shown inside the diamond.
  - The minimum cardinality is shown by the oval or hash mark next to the entity.
  - The name of the entity is shown inside the rectangle.
  - The name of the relationship is shown near the diamond.



### HAS-A Relationships

- The relationships in the previous slides are called HAS-A relationships.
- The term is used because each entity instance has a relationship to a second entity instance:
  - An employee has a badge.
  - A badge has an employee.



# Types of Entity-Relationship Diagrams

- Information Engineering (IE) [James Martin 1990]—Uses "crow's feet" to show the many sides of a relationship, and it is sometimes called the crow's foot model.
- Integrated Definition 1, Extended 3 (IDEF1X) is a version of the E-R model that is a national standard.
- Unified Modeling Language (UML) is a set of structures and techniques for modeling and designing object-oriented programs (OOP) and applications



# Crow's Foot Example: One-to-Many Relationship

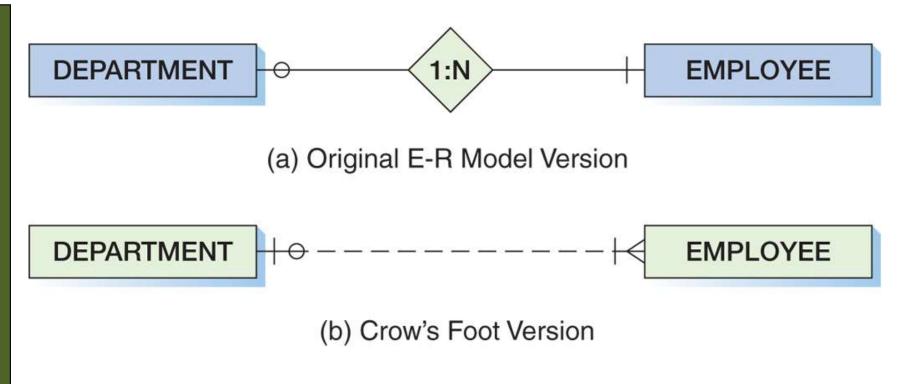


Figure 4-7: Two Versions of a 1:N Relationship



# Crow's Foot Symbols

Symbol	Meaning	Numeric Meaning
	Mandatory-One	Exactly one
	Mandatory – Many	One or more
	Optional—One	Zero or one
— <u></u>	Optional – Many	Zero or more

Figure 4-8: Crow's Foot Notation



# Crow's Foot Example: Many-to-Many Relationship

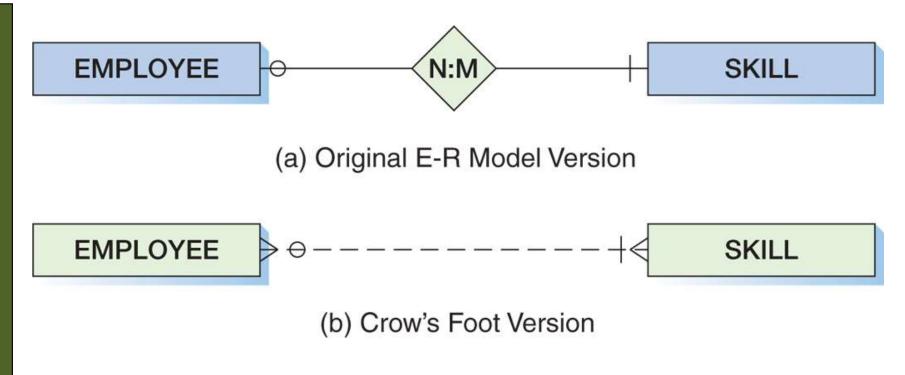


Figure 4-9: Two Versions of an N:M Relationship



# Weak Entity

- A weak entity is an entity that cannot exist in the database without the existence of another entity.
- Any entity that is not a weak entity is called a strong entity.



## ID-Dependent Weak Entities

- An ID-Dependent weak entity is a weak entity that cannot exist without its parent entity.
- An ID-dependent weak entity has a composite identifier.
  - The first part of the identifier is the identifier for the strong entity.
  - The second part of the identifier is the identifier for the weak entity itself.



# ID-Dependent Weak Entity Examples

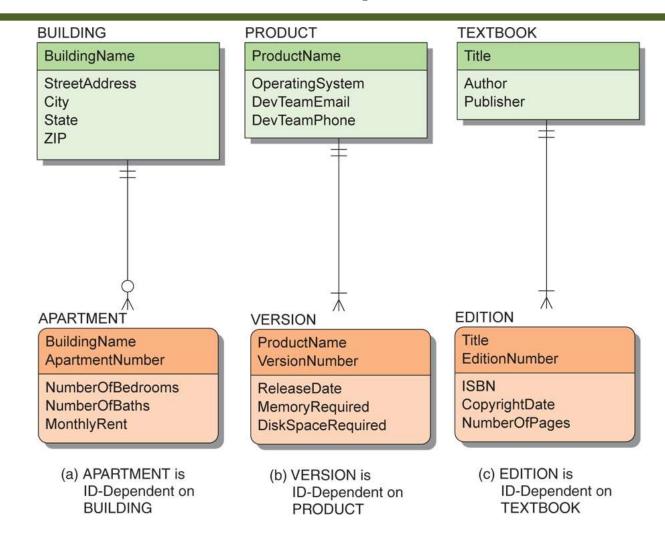


Figure 4-10: Example ID-Dependent Entities



# Weak Entity Relationships

- The relationship between a strong and weak entity is termed an identifying relationship if the weak entity is IDdependent.
  - Represented by a solid line
- The relationship between a strong and weak entity is termed a nonidentifying relationship if the weak entity is non-IDdependent.
  - Represented by a dashed line
  - Also used between strong entities



# Weak Entity Identifier: Non-ID-dependent

- All ID-dependent entities are weak entities, but there are other entities that are weak but not ID-dependent.
- A non-ID-dependent weak entity may have a single or composite identifier, but the identifier of the parent entity will be a *foreign key*.



# Non-ID-Dependent Weak Entity Examples

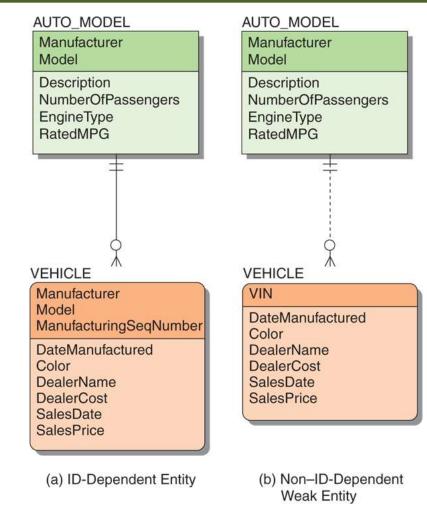


Figure 4-11: Weak Entity Examples



#### Strong and Weak Entity Examples

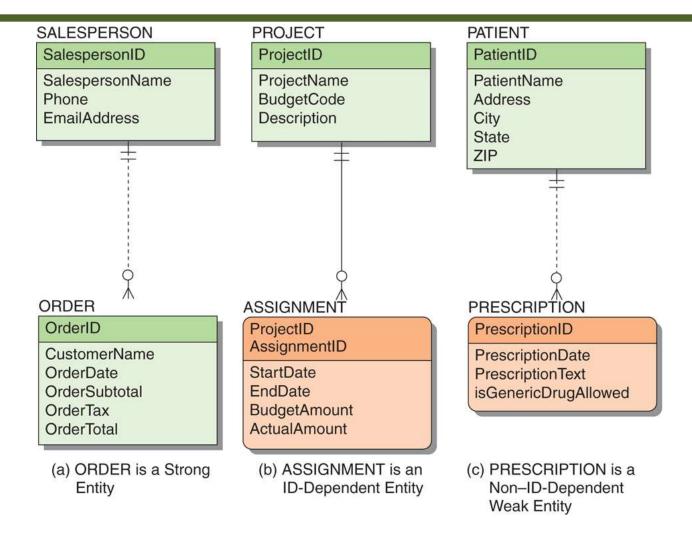


Figure 4-12: Examples of Required Entities



#### **Associative Entities**

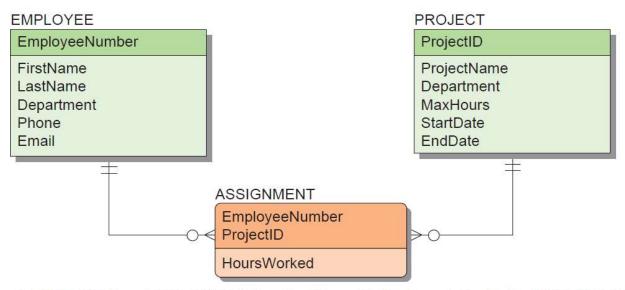
- An associative entity (also called an association entity) is used when there are attributes that are associated with the relationship between two entities rather than with either of the two entities themselves.
- A new entity is then created to:
  - Link the two original entities
  - Hold the attributes



# Associative Entities (Cont'd)



(a) N:M Relationship Between EMPLOYEE and PROJECT



(b) EMPLOYEE and PROJECT 1:N Relationships with the Associative Entity ASSIGNMENT

Figure 4-13: The Associative Entity



### Subtype Entities

- A subtype entity is a special case of another entity called supertype.
- An attribute of the supertype may be included that indicates which of the subtypes is appropriate for a given instance; this attribute is called a discriminator.
- Subtypes can be exclusive or inclusive.
  - If exclusive, the supertype relates to at most one subtype.
  - If inclusive, the supertype can relate to one or more subtypes.



### Subtype Entity Identifiers

- The relationships that connect supertypes and subtypes are called IS-A relationships because a subtype is the same entity as the supertype.
- The identifier of a supertype and all of its subtypes is the same attribute.



### Subtype Entity Examples

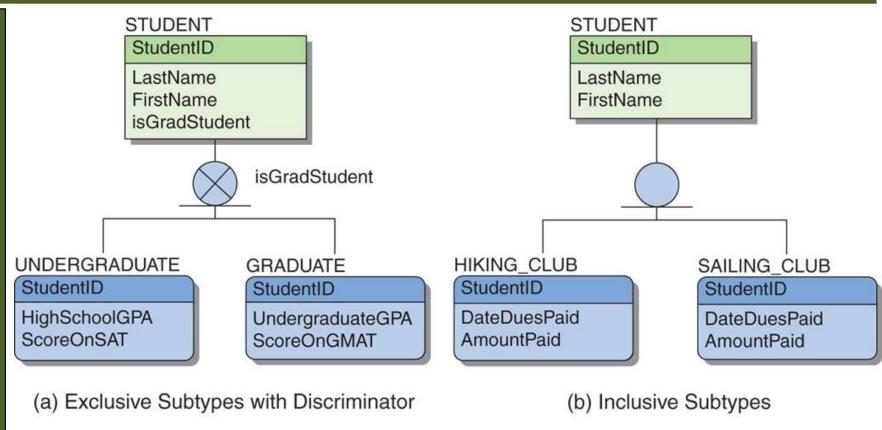


Figure 4-14: Example Subtype Entities



### Recursive Relationships

 It is possible for an entity to have a relationship to itself—this is called a recursive relationship.

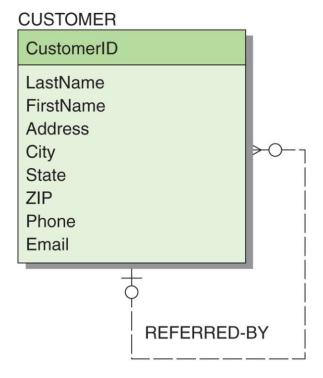


Figure 4-15: Example Recursive Relationship



### Heather Sweeney Designs

kitchen design. Her company is named Heather Sweeney Designs. Heather offers a variety of seminars at home shows, kitchen and appliance stores, and other public locations. The seminars are free; she offers them as a way of building her customer base. She earns revenue by selling books and videos that instruct people on kitchen design. She also offers custom-design consulting services. After someone attends a seminar, Heather wants to leave no stone unturned in attempting to sell that person one of her products or services. She would therefore like to develop a database to keep

track of customers, the seminars they have attended, the contacts

she has made with them, and the purchases they have made. She

wants to use this database to continue to contact her customers and

offer them products and services. The database will be named HSD.

Heather Sweeney is an interior designer who specializes in home



#### Developing an E-R Diagram

- Heather Sweeney Designs will be used as an ongoing example throughout Chapters 4, 5, 6 and 7.
  - Heather Sweeney is an interior designer who specializes in home kitchen design.
  - She offers a variety of free seminars at home shows, kitchen and appliance stores and other public locations.
  - She earns revenue by selling books and videos that instruct people on kitchen design.
  - She also offers custom-design consulting services.



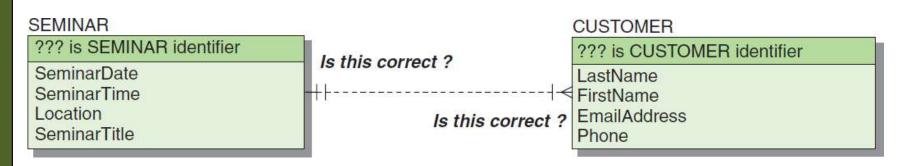
#### Heather Sweeney Designs: The Seminar Customer List

Heather Sweeney Designs Seminar Customer List							
Date: October 11, 2014 Location: San Antonio Convention Cer							
Time: 11 AM	Title: Kitchen on a Budget						
Name	Phone	Email Address					
Nancy Jacobs	817–871–8123	NJ@somewhere.com					
Chantel Jacobs	817-871-8234	CJ@somewhere.com					
Ralph Able	210-281-7687	RA@somewhere.com					
Etc.							
27 names in all							

Figure 4-16: Example Seminar Customer List



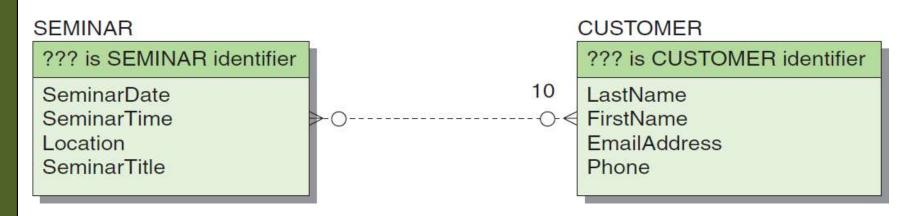
### Heather Sweeney Designs: Initial E-R Diagram I



(a) First Version of the SEMINAR and CUSTOMER E-R Diagram Figure 4-17: Initial E-R Diagram for Heather Sweeney Designs



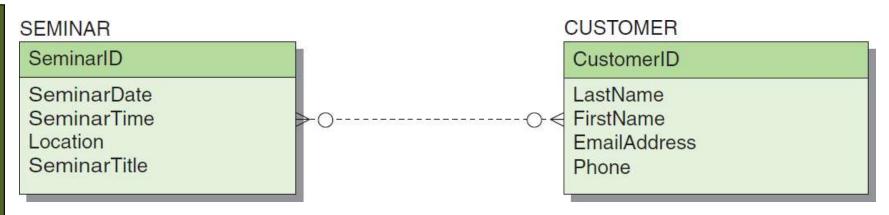
### Heather Sweeney Designs: Initial E-R Diagram II



(b) Second Version of the SEMINAR and CUSTOMER E-R Diagram Figure 4-17: Initial E-R Diagram for Heather Sweeney Designs



### Heather Sweeney Designs: Initial E-R Diagram III



(c) Third Version of the SEMINAR and CUSTOMER E-R Diagram Figure 4-17: Initial E-R Diagram for Heather Sweeney Designs



### Heather Sweeney Designs: The Customer Form Letter

#### **Heather Sweeney Designs**

122450 Rockaway Road Dallas, Texas 75227 972-233-6165

Ms. Nancy Jacobs 1400 West Palm Drive Fort Worth, Texas 76110

Dear Ms. Jacobs:

Thank you for attending my seminar "Kitchen on a Budget" at the San Antonio Convention Center. I hope that you found the seminar topic interesting and helpful for your design projects.

As a seminar attendee, you are entitled to a 15 percent discount on all of my video and book products. I am enclosing a product catalog and I would also like to invite you to visit our Web site at <a href="https://www.Sweeney.com">www.Sweeney.com</a>.

Also, as I mentioned at the seminar, I do provide customized design services to help you create that just-perfect kitchen. In fact, I have a number of clients in the Fort Worth area. Just give me a call at my personal phone number of 555-122-4873 if you'd like to schedule an appointment.

Thanks again and I look forward to hearing from you!

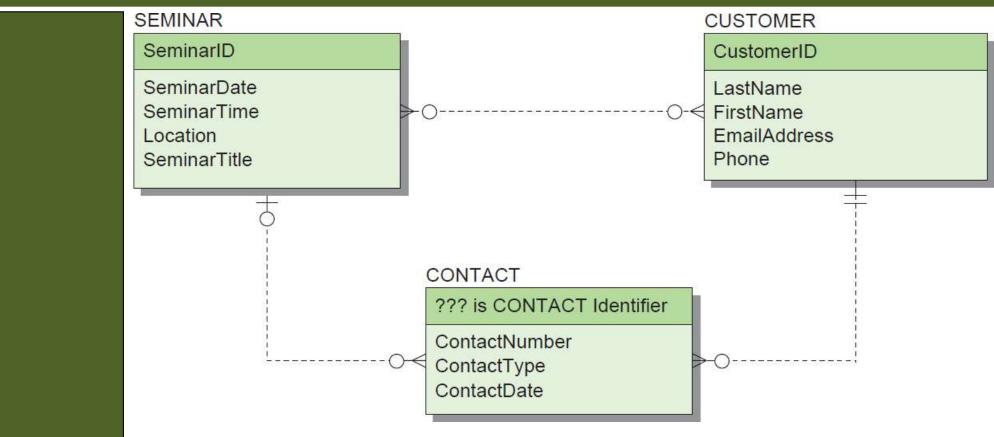
Best regards,

Heather Sweeney

Figure 4-18: Heather Sweeney Designs Customer Form Letter



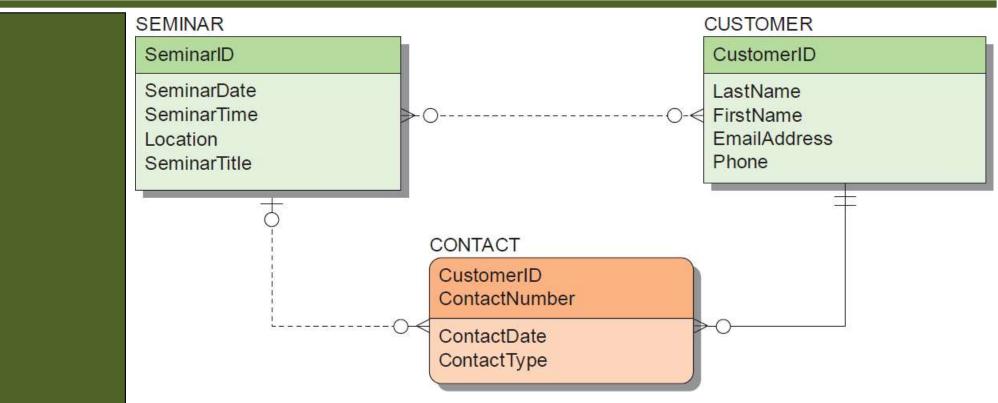
## Heather Sweeney Designs: Data Model with CONTACT



(a) First Version with CONTACT Figure 4-19: Heather Sweeney Designs Data Model with CONTACT



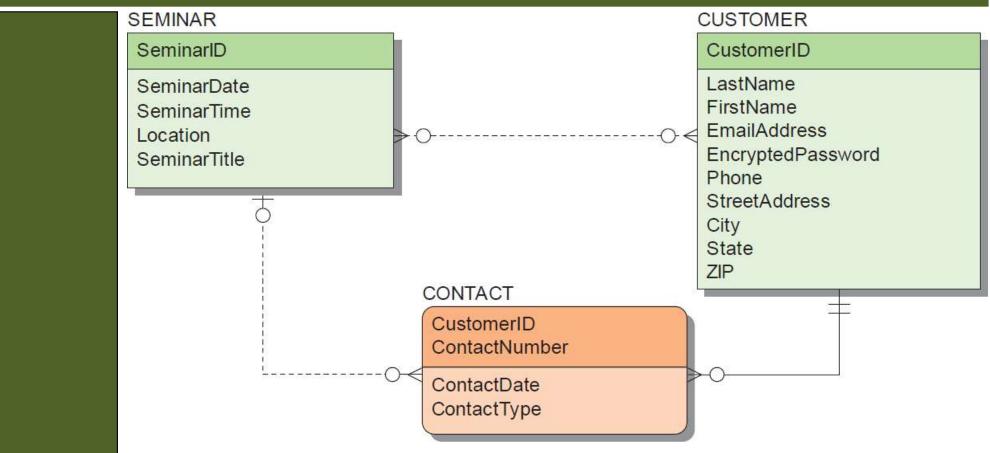
### Heather Sweeney Designs: Data Model with CONTACT as Weak Entity



(b) Second Version with CONTACT as a Weak Entity Figure 4-19: Heather Sweeney Designs Data Model with CONTACT



### Heather Sweeney Designs: Data Model with Modified CUSTOMER



(c) Third Version with Modified CUSTOMER
Figure 4-19: Heather Sweeney Designs Data Model with CONTACT



#### Heather Sweeney Designs: Sales Invoice

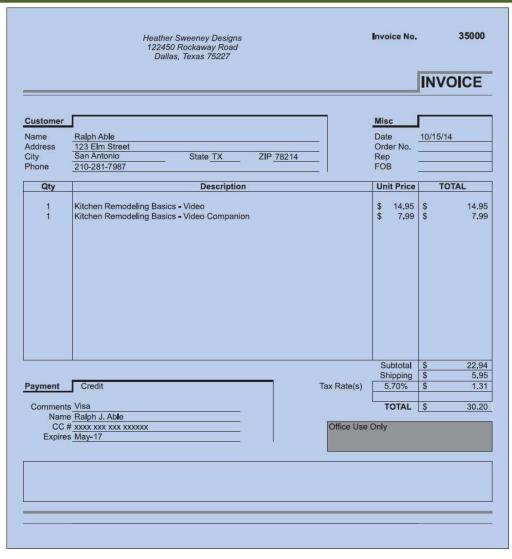


Figure 4-20: Heather Sweeney Designs Sales Invoice



#### Heather Sweeney Designs: Data Model with INVOICE

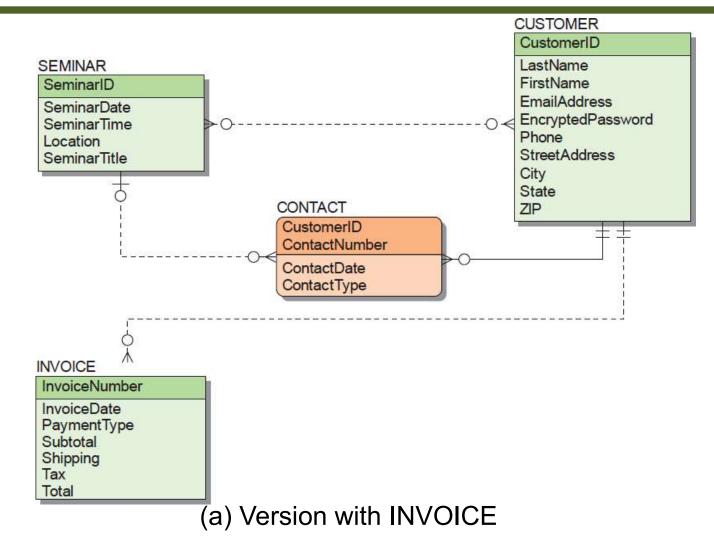
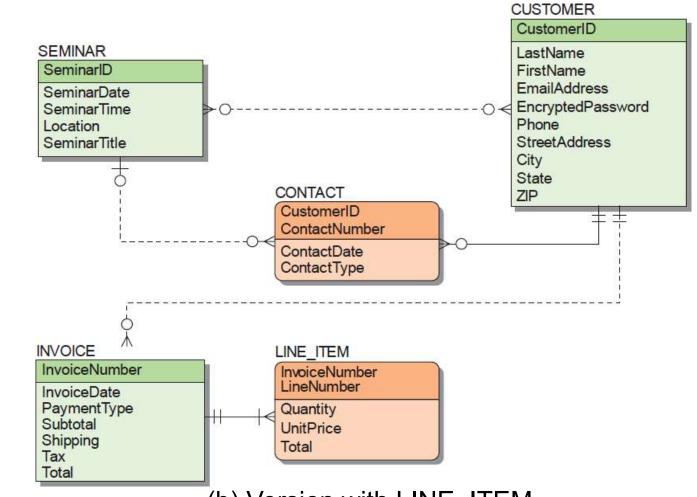


Figure 4-21: The Final Data Model for Heather Sweeney Designs



### Heather Sweeney Designs: Data Model with LINE\_ITEM

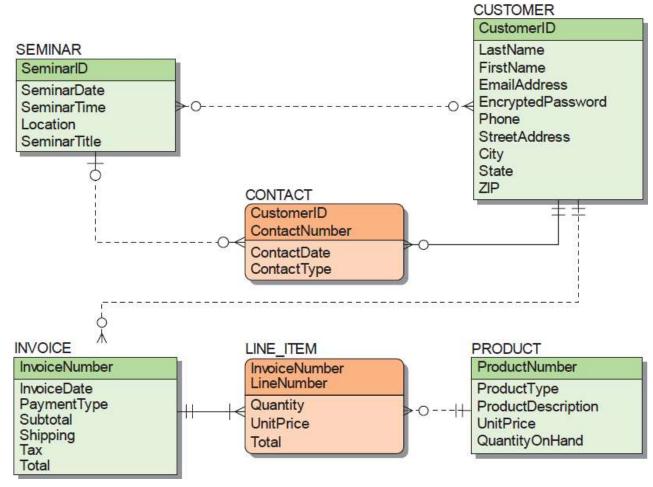


(b) Version with LINE\_ITEM

Figure 4-21: The Final Data Model for Heather Sweeney Designs



### Heather Sweeney Designs: Final Data Model



(c) The Finished Data Model

Figure 4-21: The Final Data Model for Heather Sweeney Designs



#### Heather Sweeney Designs: Business Rules and Model Validation

- Business rules may constrain the model and need to be recorded.
  - Heather Sweeney Designs has a business rule that no more than one form letter or email per day is to be sent to a customer.
- After the data model has been completed, it needs to be validated.
  - Prototyping is commonly used to validate forms and reports.



Column Name	Data Type (Length)	Key	Required	Default Value	Remarks
SeminarID	Integer	Primary Key, Foreign Key	Yes	None	REF: SEMINAR
CustomerID	Integer	Primary Key, Foreign Key	Yes	None	REF: CUSTOMER

(c) SEMINAR\_CUSTOMER

#### FIGURE 7-50

Database Column Specifications for the HSD Database

Relationship		Referential Integrity Constraint	Cascading Behavior	
Parent	Child		On Update	On Delete
SEMINAR	SEMINAR CUSTOMER	SeminarID in SEMINAR_ CUSTOMER must exist in SeminarID in SEMINAR	No	No
CUSTOMER	SEMINAR CUSTOMER	CustomerID in SEMINAR_ CUSTOMER must exist in CustomerID in CUSTOMER	No	No
SEMINAR	CONTACT	SeminarID in CONTACT must exist in SeminarID in SEMINAR	No	No
CUSTOMER	CONTACT	CustomerID in CONTACT must exist in CustomerID in CUSTOMER	No	Yes
CUSTOMER	INVOICE	CustomerID in INVOICE must exist in CustomerID in CUSTOMER	No	No
INVOICE	LINE_ITEM	InvoiceNumber in LINE_ITEM must exist in InvoiceNumber in INVOICE	No	Yes
PRODUCT	LINE_ITEM	E_ITEM ProductNumber in LINE_ITEM must exist in ProductNumber in PRODUCT		No

#### FIGURE 7-51

Referential Integrity Constraint Enforcement for the HSD Database



## DAVID M. KROENKE and DAVID J. AUER DATABASE CONCEPTS, 7<sup>th</sup> Edition

This work is protected by United States copyright laws and is provided solely for the use of instructors in teaching their courses and assessing student learning. Dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted. The work and materials from it should never be made available to students except by instructors using the accompanying text in their classes. All recipients of this work are expected to abide by these restrictions and to honor the intended pedagogical purposes and the needs of other instructors who rely on these materials.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.

Copyright © 2015 Pearson Education, Inc. Publishing as Prentice Hall



## DAVID M. KROENKE and DAVID J. AUER DATABASE CONCEPTS, 7<sup>th</sup> Edition

# End of Presentation on Chapter Four Data Modeling and the Entity-Relationship Model

