ITIS 313
Data and Information Management

Spring 2021

The Relational Database Model

A Logical View of Data

- Relational database model enables logical representation of the data and its relationships
- Logical simplicity yields simple and effective database design methodologies
- Facilitated by the creation of data relationships based on a logical construct called a relation

Table 3.1 - Characteristics of a Relational Table

1	A table is perceived as a two-dimensional structure composed of rows and columns.
2	Each table row (**tuple**) represents a single entity occurrence within the entity set.
3	Each table column represents an attribute, and each column has a distinct name.
4	Each intersection of a row and column represents a single data value.
5	All values in a column must conform to the same data format.
6	Each column has a specific range of values known as the **attribute domain**.
7	The order of the rows and columns is immaterial to the DBMS.
8	Each table must have an attribute or combination of attributes that uniquely identifies each row.

Cengage Learning © 2015

Keys

- Consist of one or more attributes that determine other attributes
- Used to:
  - Ensure that each row in a table is uniquely identifiable
  - Establish relationships among tables and to ensure the integrity of the data
- **Primary key (PK)**: Attribute or combination of attributes that uniquely identifies any given row

Determination

- State in which knowing the value of one attribute makes it possible to determine the value of another
- Is the basis for establishing the role of a key
- Based on the relationships among the attributes

Dependencies

- **Functional dependence**: Value of one or more attributes determines the value of one or more other attributes
  - **Determinant**: Attribute whose value determines another
  - **Dependent**: Attribute whose value is determined by the other attribute
- **Full functional dependence**: Entire collection of attributes in the determinant is necessary for the relationship

Types of Keys

- **Composite key**: Key that is composed of more than one attribute
- **Key attribute**: Attribute that is a part of a key
- **Entity integrity**: Condition in which each row in the table has its own unique identity
  - All of the values in the primary key must be unique
  - No key attribute in the primary key can contain a null

Types of Keys

- **Null**: Absence of any data value that could represent:
  - An unknown attribute value
  - A known, but missing, attribute value
  - A inapplicable condition
- **Referential integrity**: Every reference to an entity instance by another entity instance is valid

Table 3.3 - Relational Database Keys

KEY TYPE	DEFINITION
Superkey	An attribute or combination of attributes that uniquely identifies each row in a table
Candidate key	A minimal (irreducible) superkey; a superkey that does not contain a subset of attributes that is itself a superkey
Primary key	A candidate key selected to uniquely identify all other attribute values in any given row; cannot contain null entries
Foreign key	An attribute or combination of attributes in one table whose values must either match the primary key in another table or be null
Secondary key	An attribute or combination of attributes used strictly for data retrieval purposes

Cengage Learning © 2015

Figure 3.2 - An Example of a Simple Relational Database

Table name: PRODUCT
Primary key: PROD_CODE
Foreign key: VEND_CODE

Database name: Ch03_SaleCo

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

link

Table name: VENDOR
Primary key: VEND_CODE
Foreign key: none

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Cengage Learning © 2015

Integrity Rules

Entity Integrity	Description
Requirement	All primary key entries are unique, and no part of a primary key may be null
Purpose	Each row will have a unique identity, and foreign key values can properly reference primary key values
Example	No invoice can have a duplicate number, nor it can be null

Integrity Rules

Entity Integrity	Description
Requirement	A foreign key may have either a null entry or a entry that matches a primary key value in a table to which it is related
Purpose	It is possible for an attribute not to have a corresponding value but it is impossible to have an invalid entry It is impossible to delete row in a table whose primary keys has mandatory matching foreign key values in another table
Example	It is impossible to have invalid sales representative number

Figure 3.3 - An Illustration of Integrity Rules

Table name: CUSTOMER **Database name: Ch03_InsureCo**
Primary key: CUS_CODE
Foreign key: AGENT_CODE

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	05-Apr-2014	502
10011	Dunne	Leona	K	16-Jun-2014	501
10012	Smith	Kathy	W	29-Jan-2015	502
10013	Olowski	Paul	F	14-Oct-2014	502
10014	Orlando	Myron		28-Dec-2014	501
10015	O'Brian	Amy	B	22-Sep-2014	503
10016	Brown	James	G	25-Mar-2015	502
10017	Williams	George		17-Jul-2014	503
10018	Farriss	Anne	G	03-Dec-2014	501
10019	Smith	Olette	K	14-Mar-2015	503

Table name: AGENT (only five selected fields are shown)
Primary key: AGENT_CODE
Foreign key: none

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
501	713	228-1249	Alby	132735.75
502	615	882-1244	Hahn	138967.35
503	615	123-5589	Okon	127093.45

Cengage Learning © 2015

Ways to Handle Nulls

- **Flags**: Special codes used to indicate the absence of some value
- NOT NULL constraint - Placed on a column to ensure that every row in the table has a value for that column
- UNIQUE constraint - Restriction placed on a column to ensure that no duplicate values exist for that column

Relational Algebra

- Theoretical way of manipulating table contents using relational operators
- **Relvar**: Variable that holds a relation
  - Heading contains the names of the attributes and the body contains the relation
- Relational operators have the property of closure
  - **Closure**: Use of relational algebra operators on existing relations produces new relations

Relational Set Operators

Select (Restrict)

- Unary operator that yields a horizontal subset of a table

Project

- Unary operator that yields a vertical subset of a table

Union

- Combines all rows from two tables, excluding duplicate rows
- **Union-compatible**: Tables share the same number of columns, and their corresponding columns share compatible domains

Intersect

- Yields only the rows that appear in both tables
- Tables must be union-compatible to yield valid results

Figure 3.4 - Select

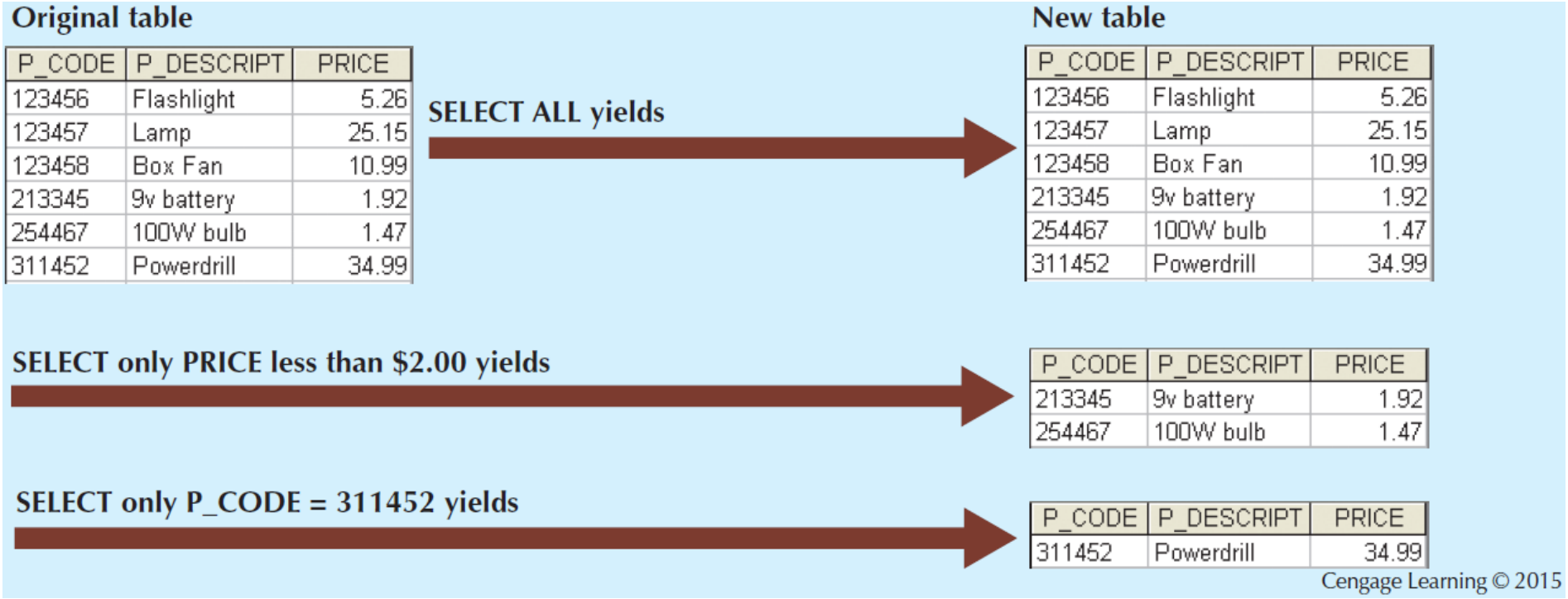


Figure 3.5 - Project

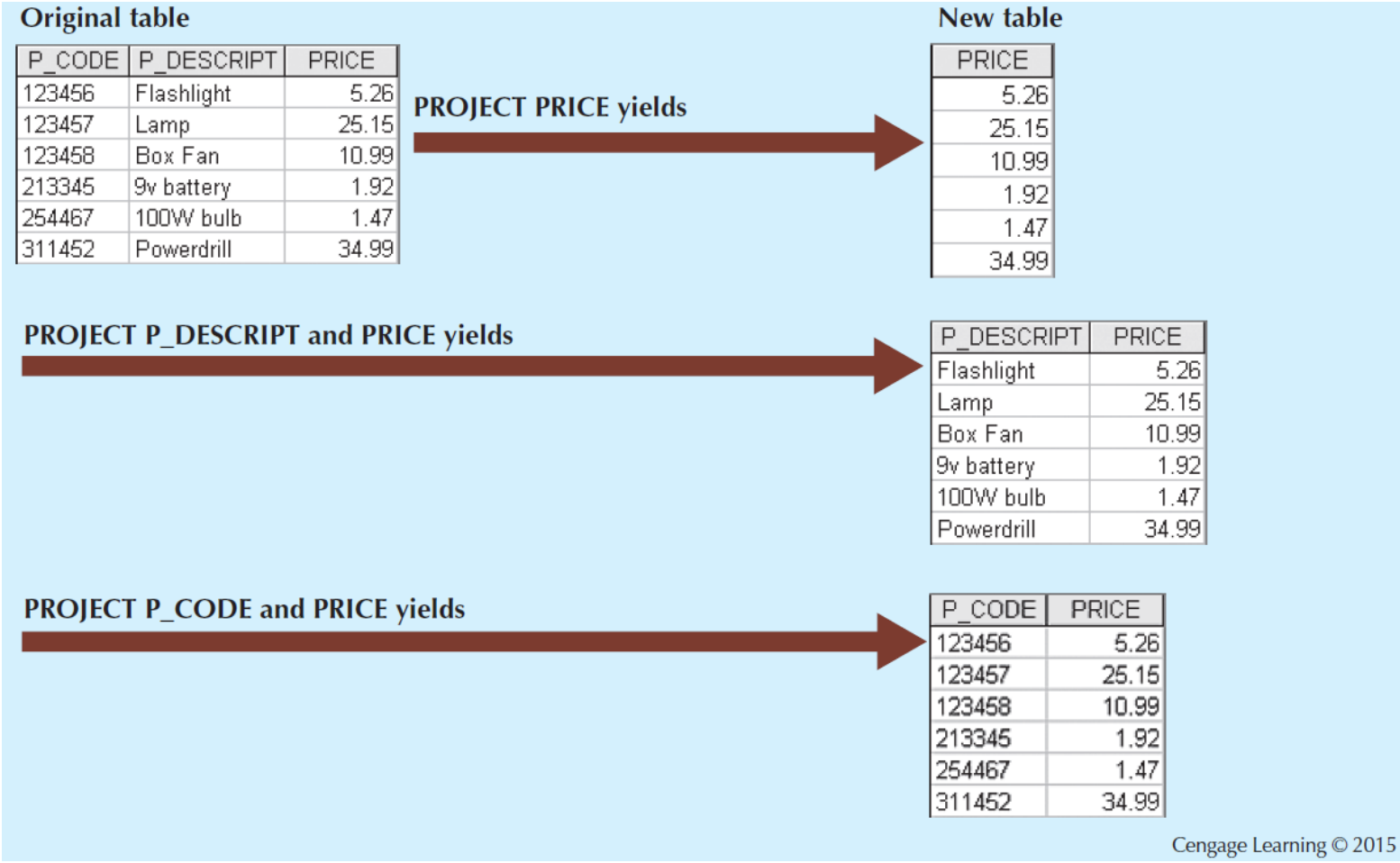


Figure 3.6 - Union

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

UNION

P_CODE	P_DESCRIPT	PRICE
345678	Microwave	160.00
345679	Dishwasher	500.00
123458	Box Fan	10.99

yields

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99
345678	Microwave	160
345679	Dishwasher	500

Cengage Learning © 2015

Figure 3.7 - Intersect

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Franklin	Johnson
Martin	Lopez

INTERSECT

EMP_FNAME	EMP_LNAME
Franklin	Lopez
William	Turner
Franklin	Johnson
Susan	Rogers

yields

STU_FNAME	STU_LNAME
Franklin	Johnson

Cengage Learning © 2015

Relational Set Operators

- **Difference**
  - Yields all rows in one table that are not found in the other table
  - Tables must be union-compatible to yield valid results
- **Product**
  - Yields all possible pairs of rows from two tables

Relational Set Operators

- **Join**
  - Allows information to be intelligently combined from two or more tables
- **Divide**
  - Uses one 2-column table as the dividend and one single-column table as the divisor
  - Output is a single column that contains all values from the second column of the dividend that are associated with every row in the divisor

Types of Joins

- **Natural join**: Links tables by selecting only the rows with common values in their common attributes
  - **Join columns**: Common columns
- **Equijoin**: Links tables on the basis of an equality condition that compares specified columns of each table
- **Theta join**: Extension of natural join, denoted by adding a theta subscript after the JOIN symbol

Types of Joins

- **Inner join**: Only returns matched records from the tables that are being joined
- **Outer join**: Matched pairs are retained and unmatched values in the other table are left null
  - **Left outer join**: Yields all of the rows in the first table, including those that do not have a matching value in the second table
  - **Right outer join**: Yields all of the rows in the second table, including those that do not have matching values in the first table

Figure 3.8 - Difference

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Franklin	Johnson
Martin	Lopez

DIFFERENCE

EMP_FNAME	EMP_LNAME
Franklin	Lopez
William	Turner
Franklin	Johnson
Susan	Rogers

yields

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Martin	Lopez

Cengage Learning © 2015

Figure 3.9 - Product

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

PRODUCT

STORE	AISLE	SHELF
23	W	5
24	K	9
25	Z	6

yields

P_CODE	P_DESCRIPT	PRICE	STORE	AISLE	SHELF
123456	Flashlight	5.26	23	W	5
123456	Flashlight	5.26	24	K	9
123456	Flashlight	5.26	25	Z	6
123457	Lamp	25.15	23	W	5
123457	Lamp	25.15	24	K	9
123457	Lamp	25.15	25	Z	6
123458	Box Fan	10.99	23	W	5
123458	Box Fan	10.99	24	K	9
123458	Box Fan	10.99	25	Z	6
213345	9v battery	1.92	23	W	5
213345	9v battery	1.92	24	K	9
213345	9v battery	1.92	25	Z	6
311452	Powerdrill	34.99	23	W	5
311452	Powerdrill	34.99	24	K	9
311452	Powerdrill	34.99	25	Z	6
254467	100W bulb	1.47	23	W	5
254467	100W bulb	1.47	24	K	9
254467	100W bulb	1.47	25	Z	6

Cengage Learning © 2015

Figure 3.10 - Two Tables That Will Be Used in JOIN Illustrations

Table name: CUSTOMER

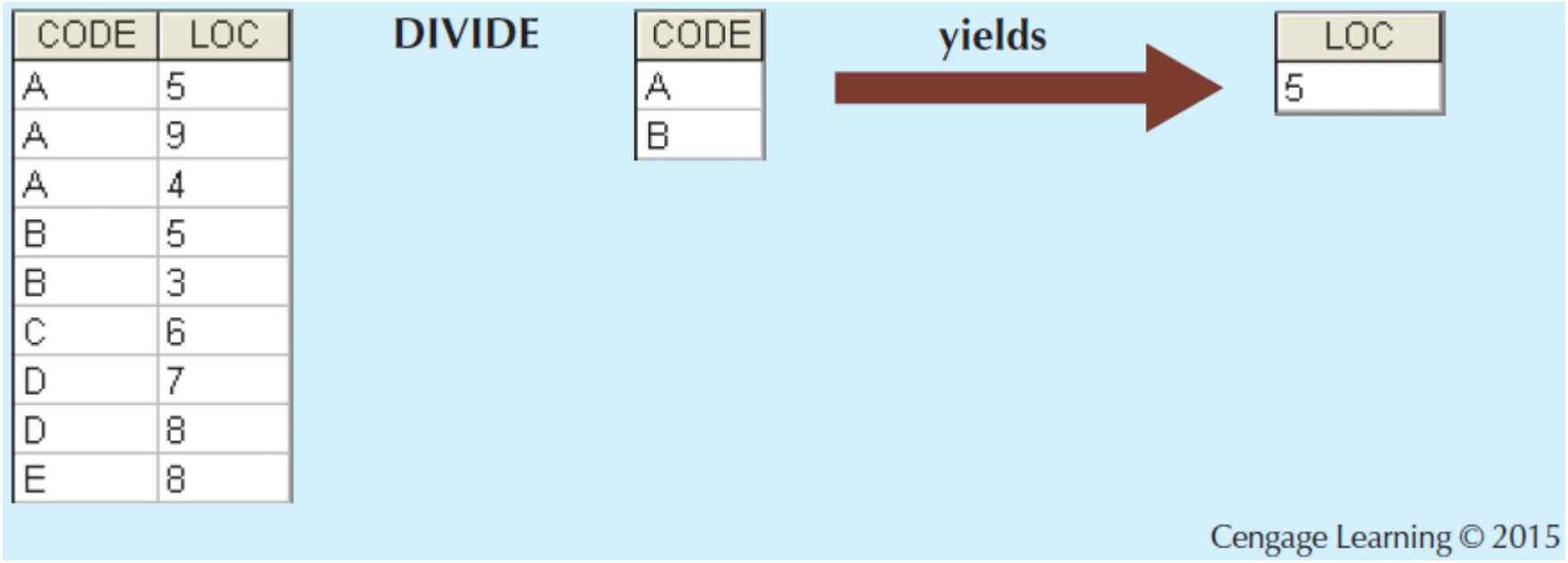
CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE
1132445	Walker	32145	231
1217782	Adares	32145	125
1312243	Rakowski	34129	167
1321242	Rodriguez	37134	125
1542311	Smithson	37134	421
1657399	Vanloo	32145	231

Table name: AGENT

AGENT_CODE	AGENT_PHONE
125	6152439887
167	6153426778
231	6152431124
333	9041234445

Cengage Learning © 2015

Figure 3.16 - Divide



Data Dictionary and the System Catalog

- **Data dictionary**: Description of all tables in the database created by the user and designer
- **System catalog**: System data dictionary that describes all objects within the database
- Homonyms and synonyms must be avoided to lessen confusion
  - **Homonym**: Same name is used to label different attributes
  - **Synonym**: Different names are used to describe the same attribute

Relationships within the Relational Database

- 1:M relationship - Norm for relational databases
- 1:1 relationship - One entity can be related to only one other entity and vice versa
- Many-to-many (M:N) relationship - Implemented by creating a new entity in 1:M relationships with the original entities
  - **Composite entity** (**Bridge** or **associative entity**): Helps avoid problems inherent to M:N relationships, includes the primary keys of tables to be linked

Figure 3.21 - The 1:1 Relationship between PROFESSOR and DEPARTMENT

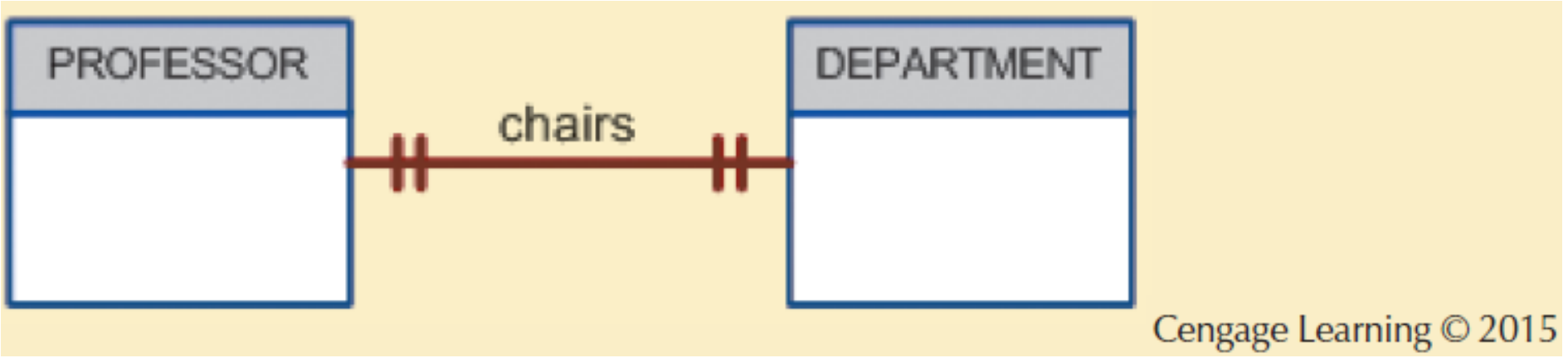


Figure 3.26 - Changing the M:N Relationship to Two 1:M Relationships

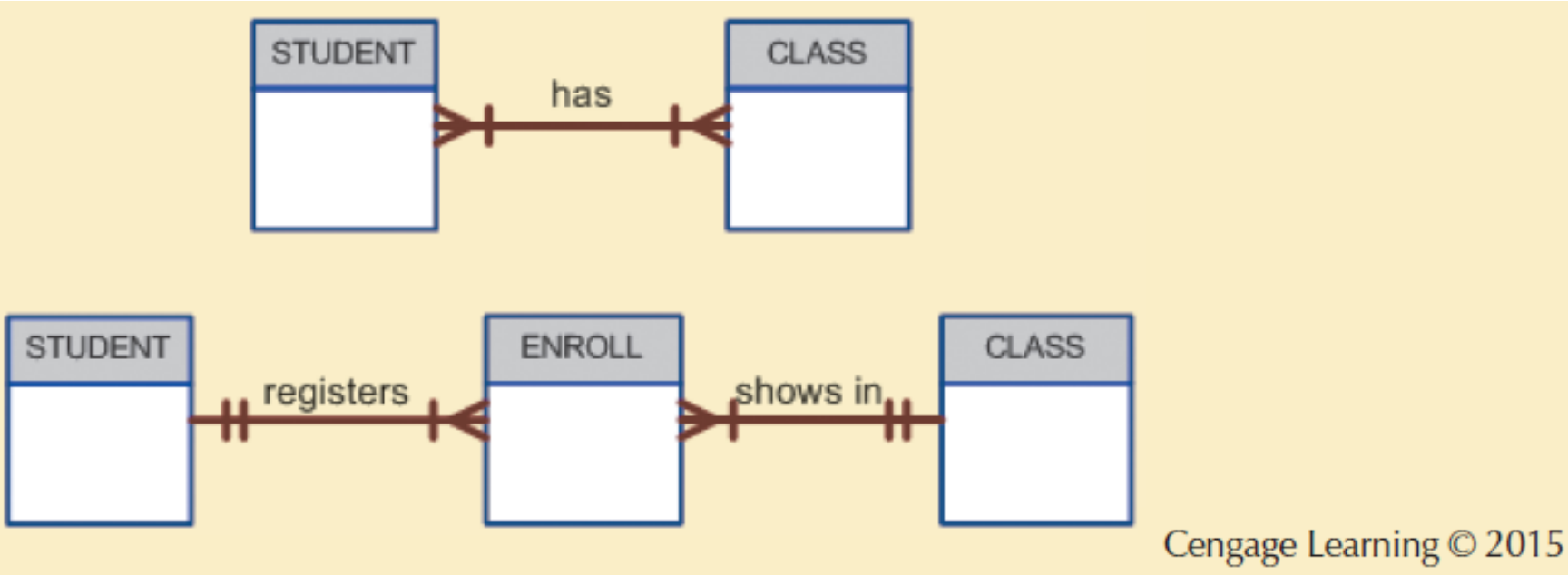
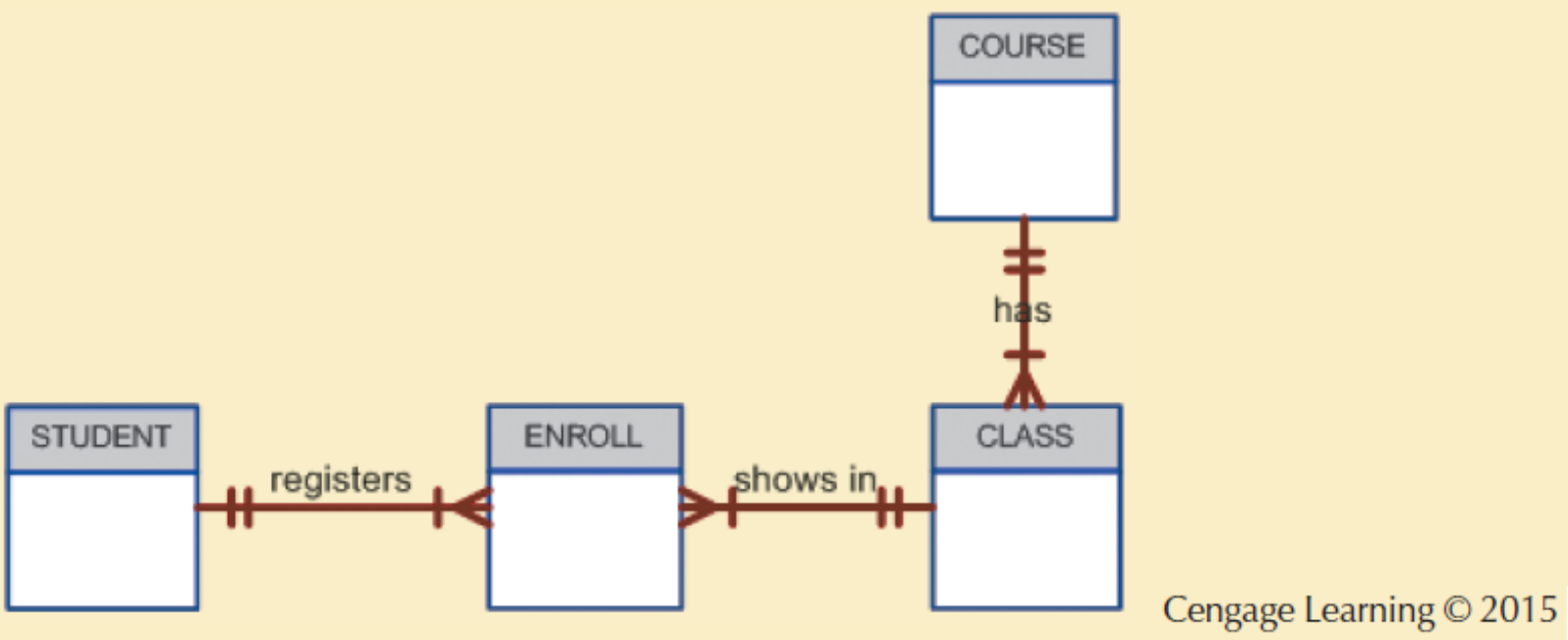


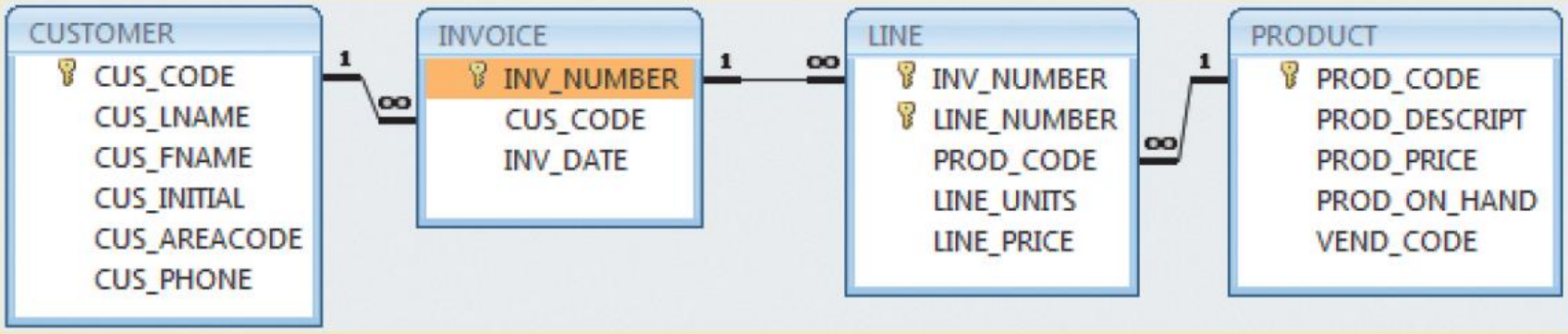
Figure 3.27 - The Expanded ER Model



Data Redundancy

- Relational database facilitates control of data redundancies through use of foreign keys
- To be controlled except the following circumstances
  - Data redundancy must be increased to make the database serve crucial information purposes
  - Exists to preserve the historical accuracy of the data

Figure 3.30 - The Relational Diagram for the Invoicing System



Cengage Learning © 2015

Index

- Orderly arrangement to logically access rows in a table
- **Index key**: Index's reference point that leads to data location identified by the key
- **Unique index**: Index key can have only one pointer value associated with it
- Each index is associated with only one table