3

Chapter 3

The Relational Database Model

Database Systems:
Design, Implementation, and Management,
Sixth Edition, Rob and Coronel

In this chapter, you will learn:

- That the relational database model takes a logical view of data
- That the relational model's basic components are entities, attributes, and relationships among entities
- How entities and their attributes are organized into tables

In this chapter, you will learn (continued):

- About relational database operators, the data dictionary, and the system catalog
- How data redundancy is handled in the relational database model
- Why indexing is important

A Logical View of Data

- Relational model
  - Enables us to view data *logically* rather than *physically*
  - Reminds us of simpler file concept of data storage
- Table
  - Has advantages of structural and data independence
  - Resembles a file from conceptual point of view
  - Easier to understand than its hierarchical and network database predecessors

Tables and Their Characteristics

- **Table**: two-dimensional structure composed of rows and columns
- Contains group of related entities→ an entity set
  - Terms *entity set* and *table* are often used interchangeably

Tables and Their Characteristics (continued)

- Table also called a *relation* because the relational model's creator, Codd, used the term *relation* as a synonym for table
- Think of a table as a *persistent relation*:
  - A relation whose contents can be permanently saved for future use

Characteristics of a Relational Table

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 row/column intersection represents a single data value.

TABLE 3.1 CHARACTERISTICS OF A RELATIONAL TABLE (CONTINUED)

5	All values in a column must conform to the same data format. For example, if the attribute is assigned an integer data format, all values in the column representing that attribute must be integers.
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 a combination of attributes that uniquely identifies each row.

STUDENT Table Attribute Values

FIGURE 3.1 STUDENT TABLE ATTRIBUTE VALUES

Database name: Ch03_TinyCollege
Table name: STUDENT

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS
321452	Bowser	William	C	12-Feb-1972	42	So
324257	Smithson	Anne	K	15-Nov-1977	81	Jr
324258	Brewer	Juliette		23-Aug-1966	36	So
324269	Oblonski	Walter	H	16-Sep-1973	66	Jr
324273	Smith	John	D	30-Dec-1955	102	Sr
324274	Katinga	Raphael	P	21-Oct-1976	114	Sr
324291	Robertson	Gerald	T	08-Apr-1970	120	Sr
324299	Smith	John	B	30-Nov-1983	15	Fr

STUDENT table, continued

STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
2.84	No	BIOL	2134	205
3.27	Yes	CIS	2256	222
2.26	Yes	ACCT	2256	228
3.09	No	CIS	2114	222
2.11	Yes	ENGL	2231	199
3.15	No	ACCT	2267	228
3.87	No	EDU	2267	311
2.92	No	ACCT	2315	230

STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_DOB = Student date of birth
STU_GPA = Grade point average
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

Keys

- Consists of one or more attributes that determine other attributes
- Primary key (PK) is an attribute (or a combination of attributes) that uniquely identifies any given entity (row)
- Key's role is based on determination
  - If you know the value of attribute A, you can look up (determine) the value of attribute B

Student Classification

TABLE 3.2 STUDENT CLASSIFICATION

HOURS COMPLETED	CLASSIFICATION
Less then 30	Fr
30–59	So
60–89	Jr
90 or more	Sr

Keys (continued)

- Composite key
  - Composed of more than one attribute
- Key attribute
  - Any attribute that is part of a key
- Superkey
  - Any key that uniquely identifies each entity
- Candidate key
  - A superkey without redundancies

Null Values

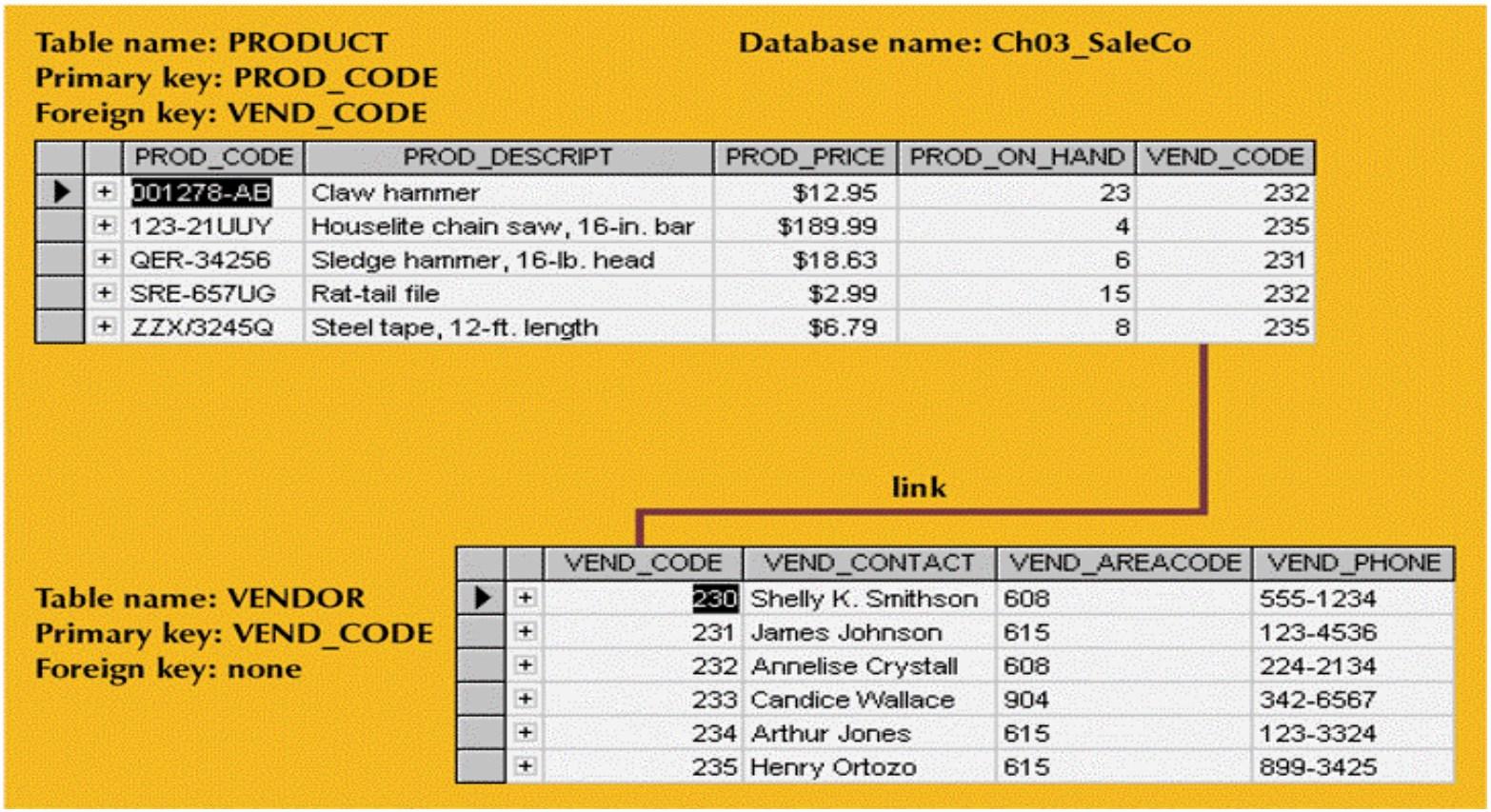
- No data entry
- Not permitted in primary key
- Should be avoided in other attributes
- Can represent
  - An unknown attribute value
  - A known, but missing, attribute value
  - A “not applicable” condition
- Can create problems in logic and using formulas

Controlled Redundancy

- Makes the relational database work
- Tables within the database share common attributes that enable us to link tables together
- Multiple occurrences of values in a table are not redundant when they are *required* to make the relationship work
- Redundancy is *unnecessary* duplication of data

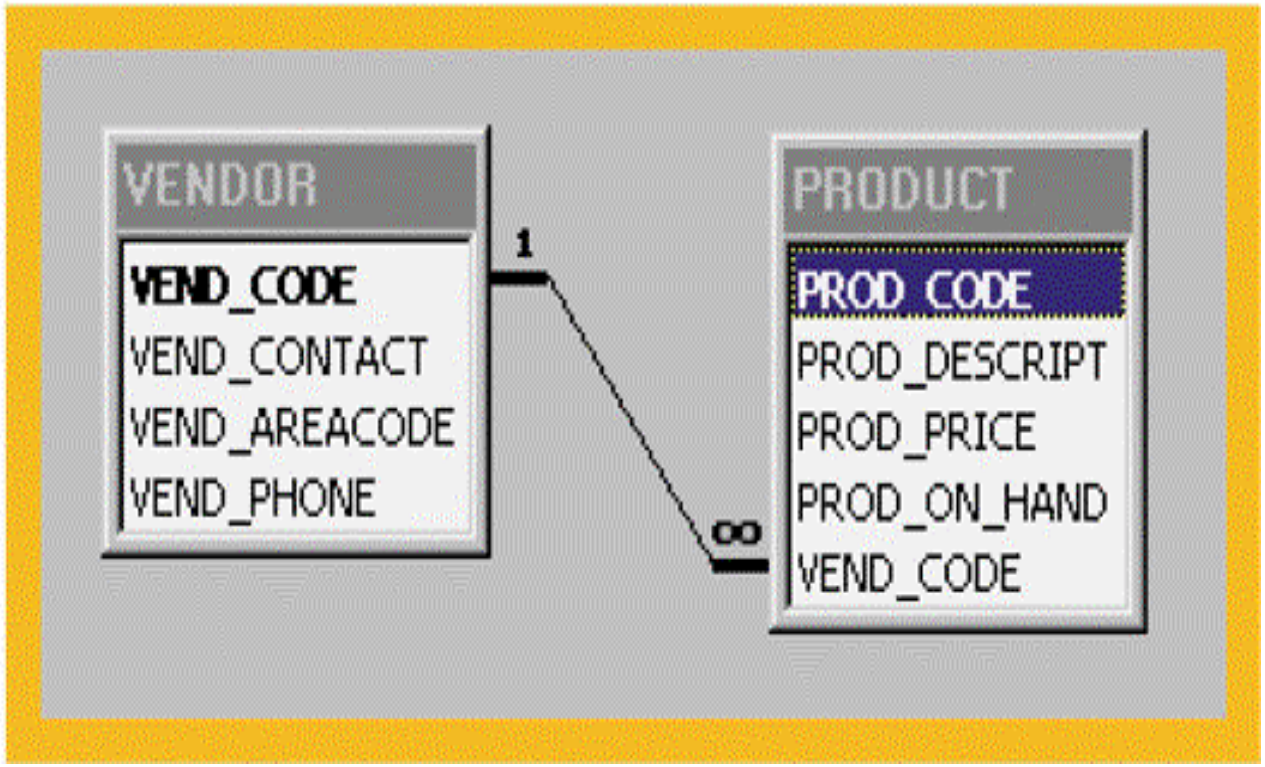
An Example of a Simple Relational Database

FIGURE 3.2 AN EXAMPLE OF A SIMPLE RELATIONAL DATABASE



The Relational Schema for the CH03_SaleCo Database

FIGURE 3.3 THE RELATIONAL SCHEMA FOR THE CH03_SALECO DATABASE



Keys (continued)

- Foreign key (FK)
  - An attribute whose values match primary key values in the related table
- Referential integrity
  - FK contains a value that refers to an existing valid tuple (row) in another relation
- Secondary key
  - Key used strictly for data retrieval purposes

Relational Database Keys

TABLE 3.3 RELATIONAL DATABASE KEYS

KEY TYPE	DEFINITION
Superkey	An attribute (or combination of attributes) that uniquely identifies each entity in a table.
Candidate key	A minimal 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.
Secondary key	An attribute (or combination of attributes) used strictly for data retrieval purposes.
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.

Integrity Rules

TABLE 3.4 INTEGRITY RULES

ENTITY INTEGRITY	DESCRIPTION
Requirement	All primary key entries are unique, and no part of a primary key may be null.
Purpose	Guarantees that each entity will have a unique identity and ensures that foreign key values can properly reference primary key values.
Example	No invoice can have a duplicate number, nor can it be null. In short, all invoices are uniquely identified by their invoice number.
REFERENTIAL INTEGRITY	**DESCRIPTION**
Requirement	A foreign key may have either a null entry—as long as it is not a part of its table's primary key—or an entry that matches the primary key value in a table to which it is related. (Every non-null foreign key value *must* reference an *existing* primary key value.)
Purpose	Makes it possible for an attribute NOT to have a corresponding value, but it will be impossible to have an invalid entry. The enforcement of the referential integrity rule makes it impossible to delete a row in one table whose primary key has mandatory matching foreign key values in another table.
Example	A customer might not (yet) have an assigned sales representative (number), but it will be impossible to have an invalid sales representative (number).

An Illustration of Integrity Rules

FIGURE 3.4 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_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	12-Mar-02	502
10011	Dunne	Leona	K	713	894-1238	23-May-02	501
10012	Smith	Kathy	W	615	894-2285	05-Jan-03	502
10013	Olowski	Paul	F	615	894-2180	20-Sep-02	
10014	Orlando	Myron		615	222-1672	04-Dec-02	501
10015	O'Brian	Amy	B	713	442-3381	29-Aug-02	503
10016	Brown	James	G	615	297-1228	01-Mar-03	502
10017	Williams	George		615	290-2556	23-Jun-02	503
10018	Farriss	Anne	G	713	382-7185	09-Nov-02	501
10019	Smith	Olette	K	615	297-3809	18-Feb-03	503

Table name: AGENT
Primary key: AGENT_CODE
Foreign key: none

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
501	713	228-1249	Alby	$1,735,453.75
502	615	882-1244	Hahn	$4,967,003.28
503	615	123-5589	Okon	$3,093,980.41

A Dummy Variable Value Used as a Flag

TABLE 3.5 A DUMMY VARIABLE VALUE USED AS A FLAG

AGENT_CODE	AGENT_AREA CODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SALES
-99	000	000-0000	None	$0.00

Relational Database Operators

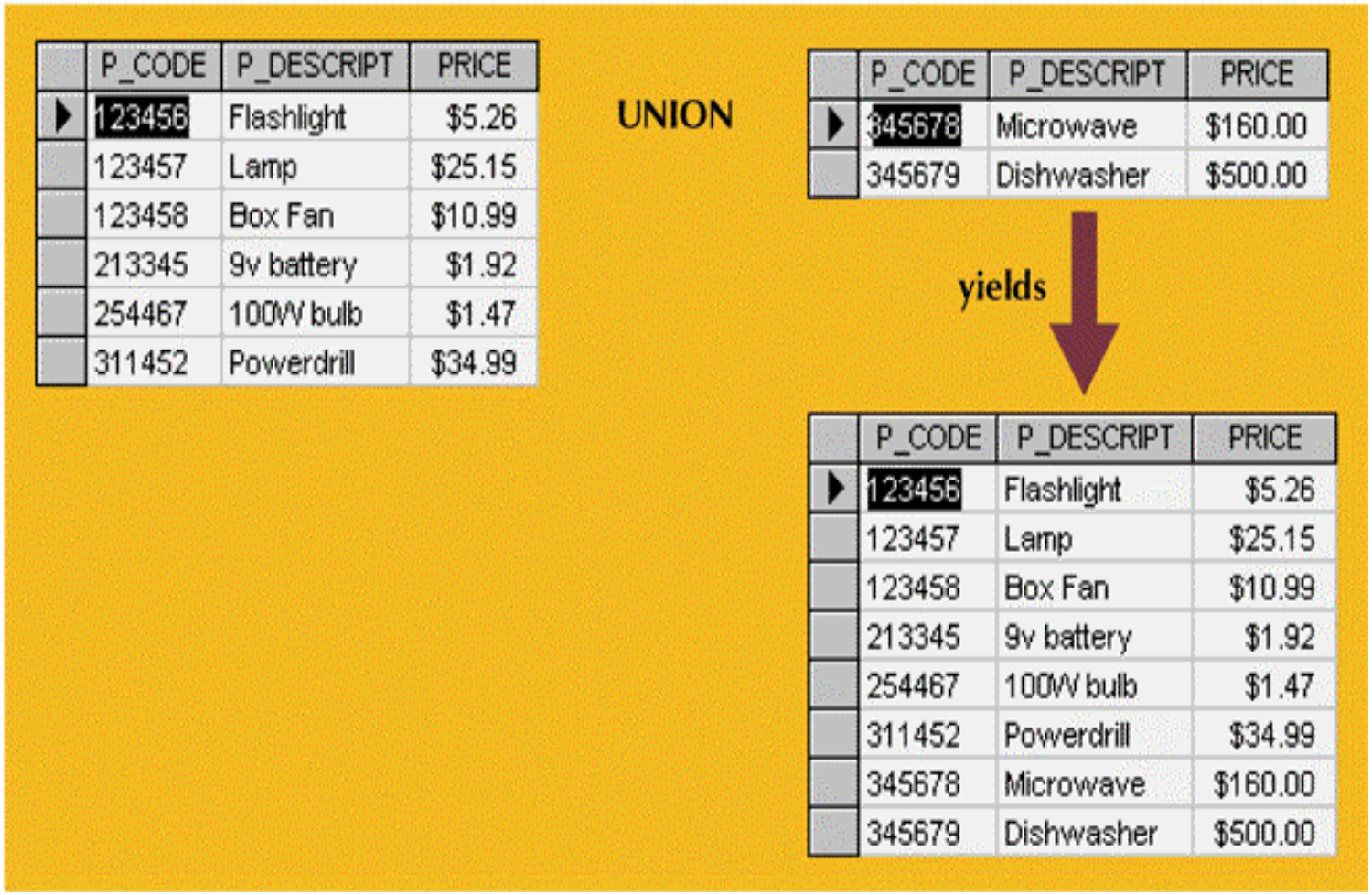
- **Relational algebra**
  - Defines theoretical way of manipulating table contents using relational operators:
    - SELECT
    - PROJECT
    - JOIN
    - INTERSECT
    - UNION
    - DIFFERENCE
    - PRODUCT
    - DIVIDE
  - Use of relational algebra operators on existing tables (relations) produces new relations

Relational Algebra Operators (continued)

- Union:
  - Combines all rows from two tables, excluding duplicate rows
  - Tables must have the same attribute characteristics
- Intersect:
  - Yields only the rows that appear in both tables

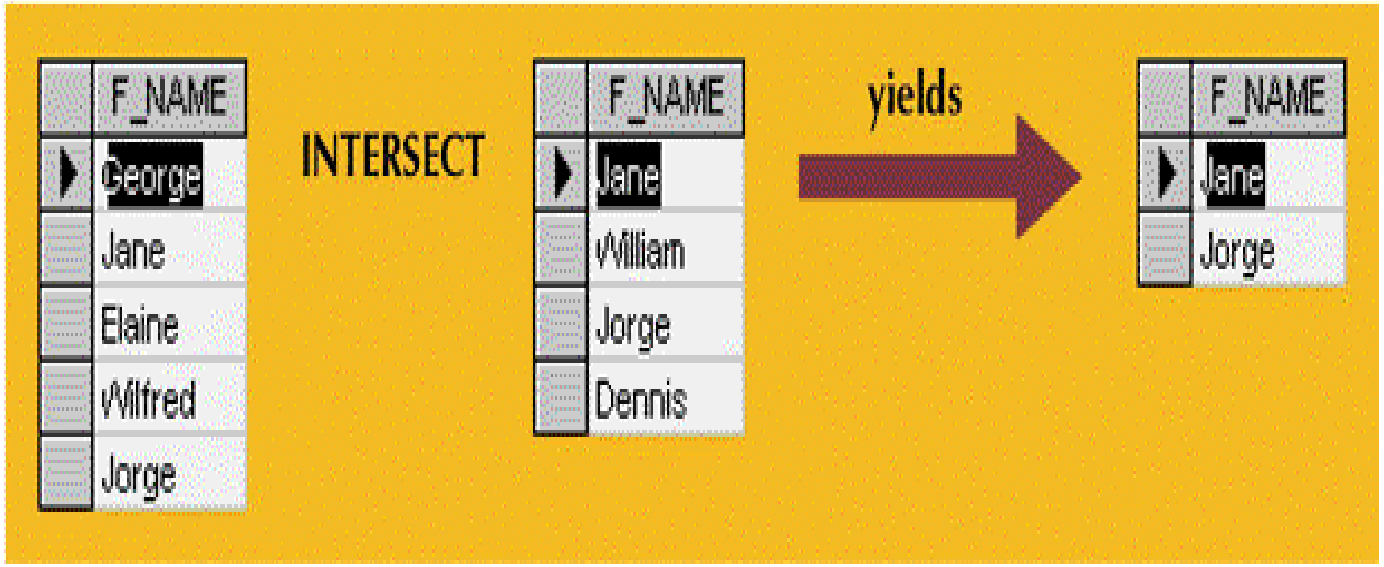
Union

FIGURE 3.5 UNION



Intersect

FIGURE 3.6 INTERSECT

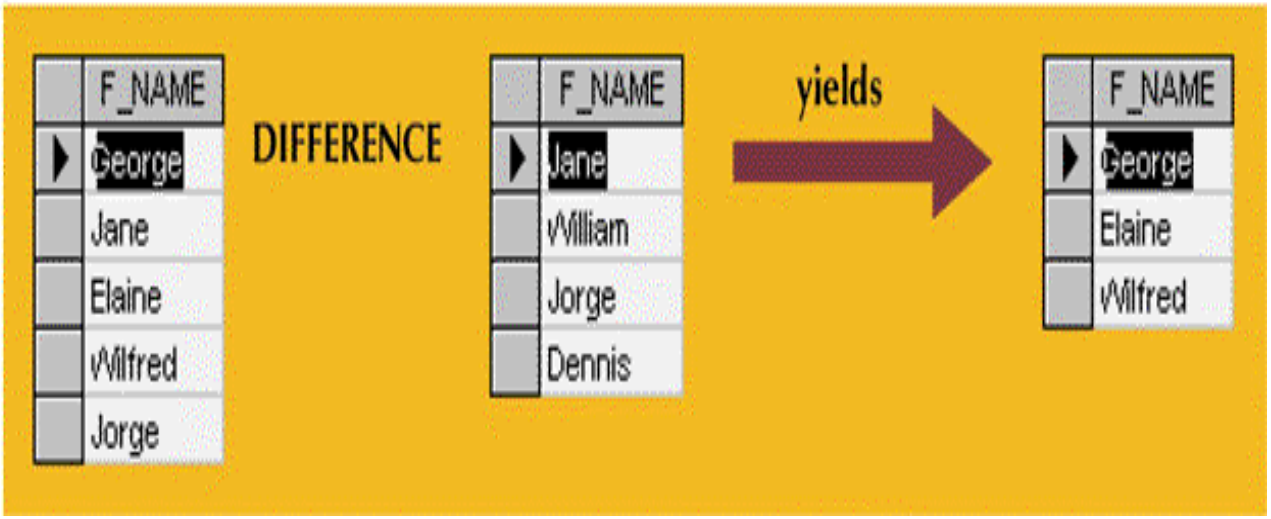


Relational Algebra Operators (continued)

- Difference
  - Yields all rows in one table not found in the other table—that is, it subtracts one table from the other
- Product
  - Yields all possible pairs of rows from two tables
    - Also known as the Cartesian product

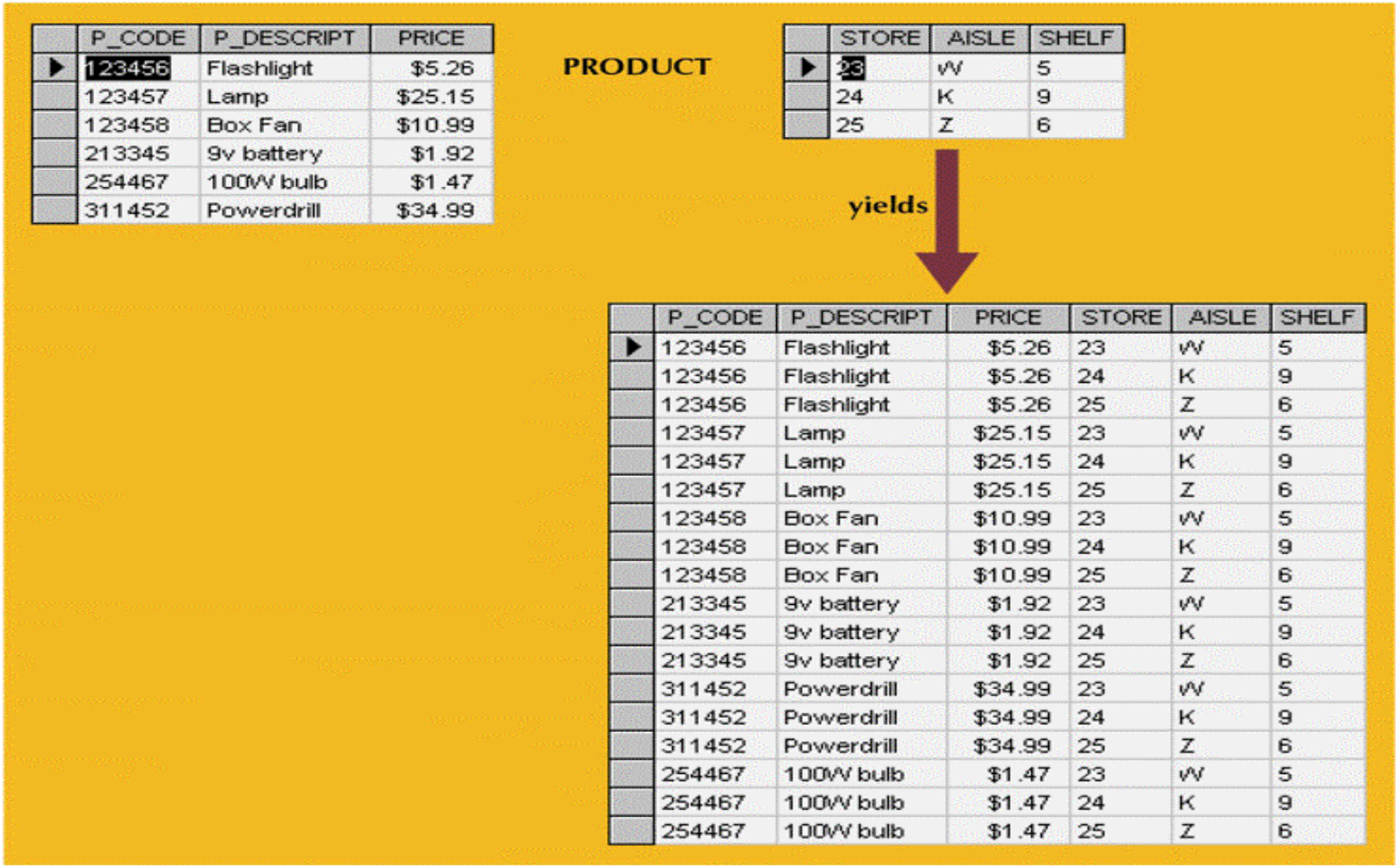
Difference

FIGURE 3.7 DIFFERENCE



Product

FIGURE 3.8 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

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

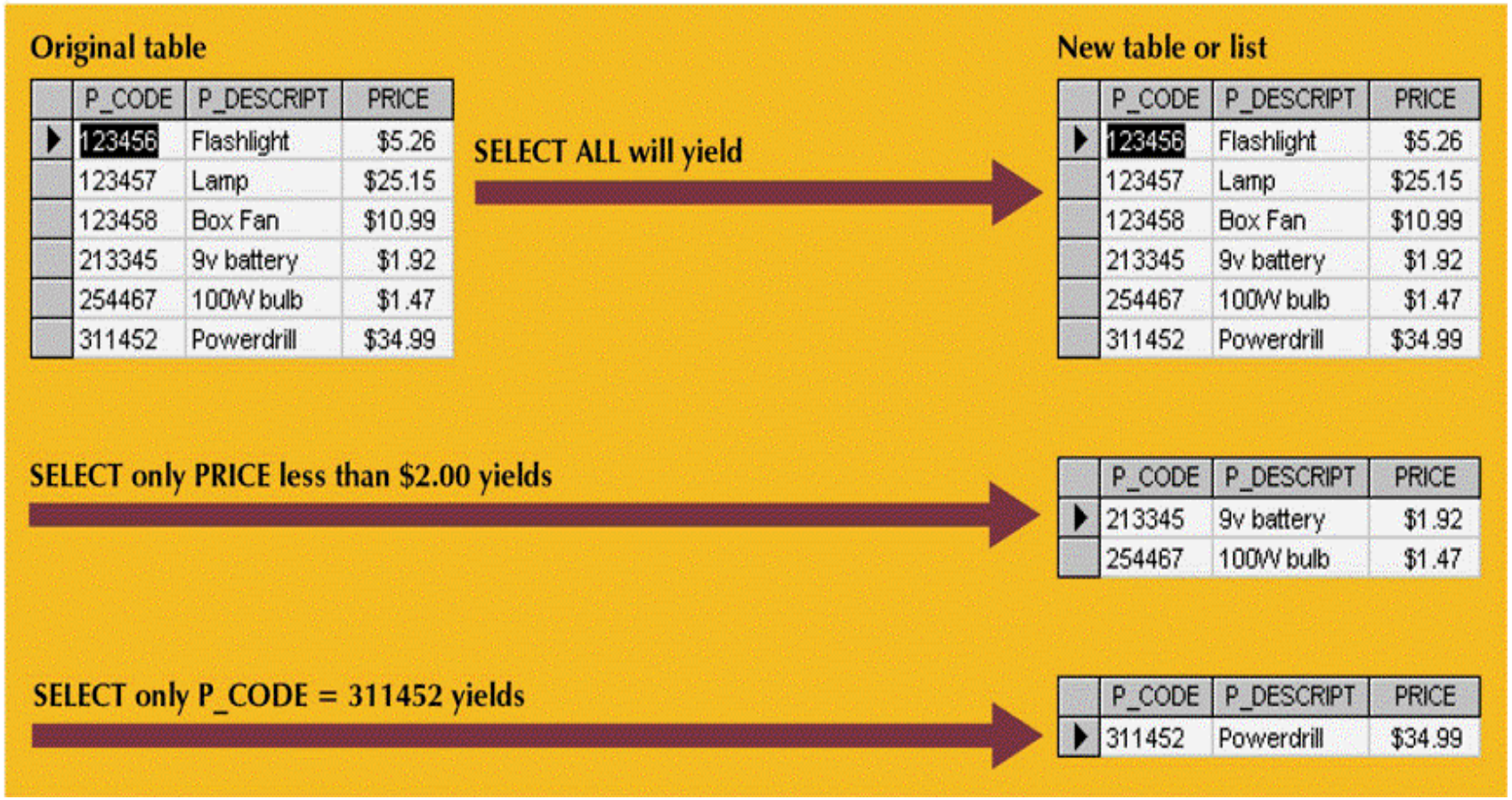
	P_CODE	P_DESCRIPT	PRICE	STORE	AISLE	SHELF
▶	123456	Flashlight	$5.26	23	W	5
	123456	Flashlight	$5.26	24	K	9
	123456	Flashlight	$5.26	25	Z	6
	123457	Lamp	$25.15	23	W	5
	123457	Lamp	$25.15	24	K	9
	123457	Lamp	$25.15	25	Z	6
	123458	Box Fan	$10.99	23	W	5
	123458	Box Fan	$10.99	24	K	9
	123458	Box Fan	$10.99	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

Relational Algebra Operators (continued)

- Select
  - Yields values for all rows found in a table
  - Can be used to list either all row values or it can yield only those row values that match a specified criterion
  - Yields a horizontal subset of a table
- Project
  - Yields all values for selected attributes
  - Yields a vertical subset of a table

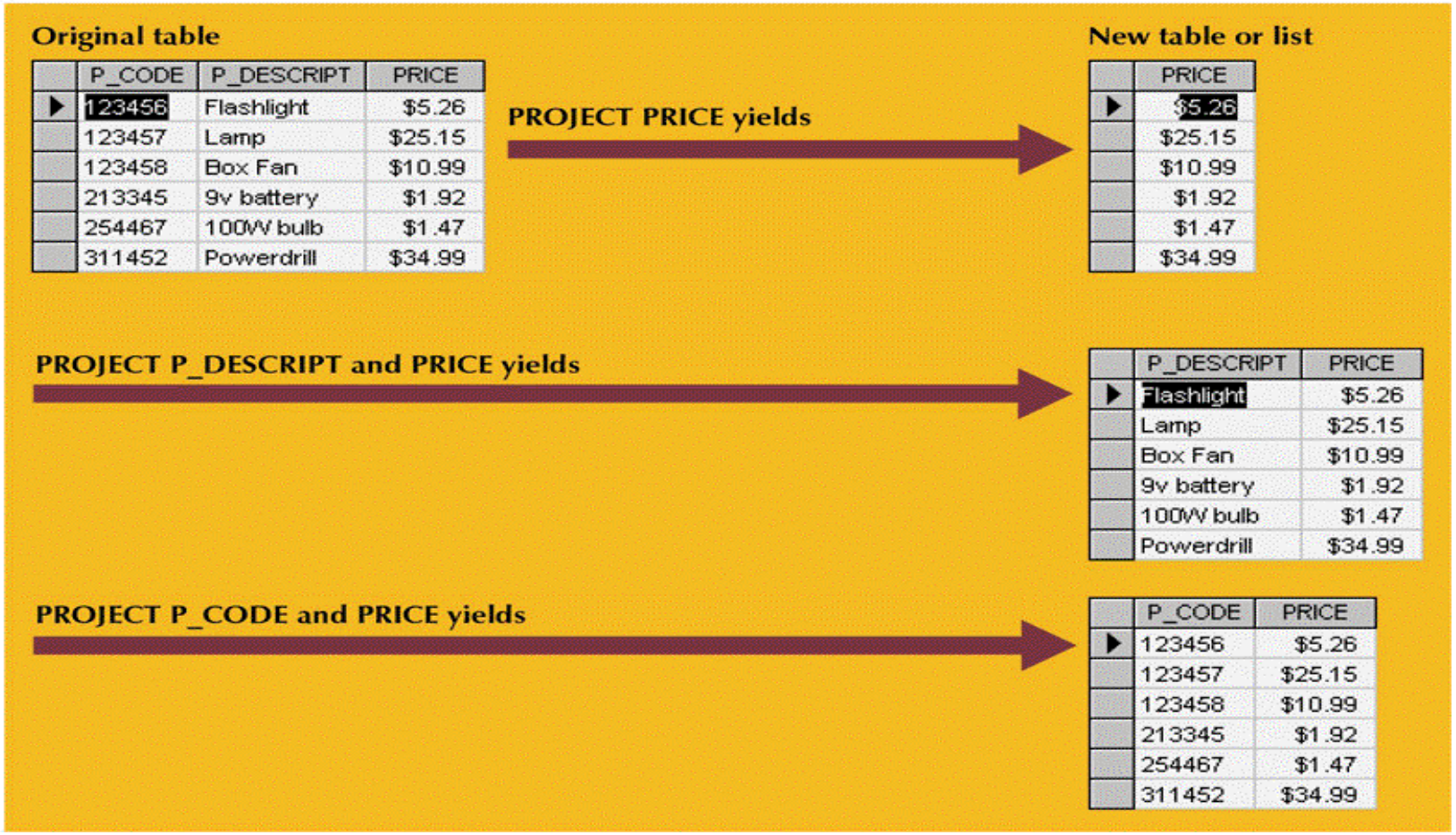
Select

FIGURE 3.9 **SELECT**



Project

FIGURE 3.10 PROJECT



Relational Algebra Operators (continued)

- Join
  - Allows us to combine information from two or more tables
  - Real power behind the relational database, allowing the use of independent tables linked by common attributes

Two Tables That Will Be Used in Join Illustrations

FIGURE 3.11 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

Natural Join

- Links tables by selecting only rows with common values in their common attribute(s)
- Result of a three-stage process:
  1. PRODUCT of the tables is created
  2. SELECT is performed on Step 1 output to yield only the rows for which the AGENT_CODE values are equal
     - Common column(s) are called join column(s)
  3. PROJECT is performed on Step 2 results to yield a single copy of each attribute, thereby eliminating duplicate columns

Natural Join, Step 1: PRODUCT

FIGURE 3.12 NATURAL JOIN, STEP 1: PRODUCT

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1132445	Walker	32145	231	125	6152439887
1132445	Walker	32145	231	167	6153426778
1132445	Walker	32145	231	231	6152431124
1132445	Walker	32145	231	333	9041234445
1217782	Adares	32145	125	125	6152439887
1217782	Adares	32145	125	167	6153426778
1217782	Adares	32145	125	231	6152431124
1217782	Adares	32145	125	333	9041234445
1312243	Rakowski	34129	167	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1312243	Rakowski	34129	167	231	6152431124
1312243	Rakowski	34129	167	333	9041234445
1321242	Rodriguez	37134	125	125	6152439887
1321242	Rodriguez	37134	125	167	6153426778
1321242	Rodriguez	37134	125	231	6152431124
1321242	Rodriguez	37134	125	333	9041234445
1542311	Smithson	37134	421	125	6152439887
1542311	Smithson	37134	421	167	6153426778
1542311	Smithson	37134	421	231	6152431124
1542311	Smithson	37134	421	333	9041234445
1657399	Vanloo	32145	231	125	6152439887
1657399	Vanloo	32145	231	167	6153426778
1657399	Vanloo	32145	231	231	6152431124
1657399	Vanloo	32145	231	333	9041234445

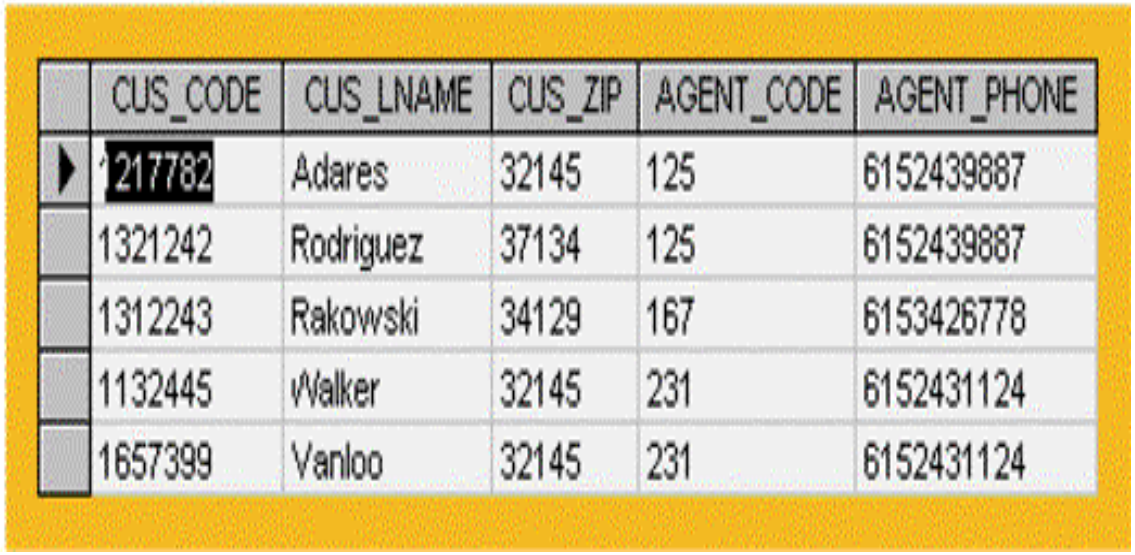
Natural Join, Step 2: SELECT

FIGURE 3.13 NATURAL JOIN, STEP 2: SELECT

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	125	6152439887
1321242	Rodriguez	37134	125	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1132445	Walker	32145	231	231	6152431124
1657399	Vanloo	32145	231	231	6152431124

Natural Join, Step 3: PROJECT

FIGURE 3.14 NATURAL JOIN, STEP 3: PROJECT



CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124

Natural Join (continued)

- Final outcome yields table that
  - Does not include unmatched pairs
  - Provides only copies of matches
- If no match is made between the table rows,
  - the new table does not include the unmatched row

Natural Join (continued)

- The column on which we made the JOIN—that is, AGENT_CODE—occurs only once in the new table
- If the same AGENT_CODE were to occur several times in the AGENT table,
  - a customer would be listed for each match

Other Forms of Join

- Equijoin
  - Links tables on the basis of an equality condition that compares specified columns of each table
  - Outcome does not eliminate duplicate columns
  - Condition or criterion to join tables must be explicitly defined
  - Takes its name from the equality comparison operator (=) used in the condition
- Theta join
  - If any other comparison operator is used

Outer Join

- Matched pairs are retained and any unmatched values in other table are left null
- In outer join for tables CUSTOMER and AGENT, two scenarios are possible:
  - Left outer join
    - Yields all rows in CUSTOMER table, including those that do not have a matching value in the AGENT table
  - Right outer join
    - Yields all rows in AGENT table, including those that do not have matching values in the CUSTOMER table

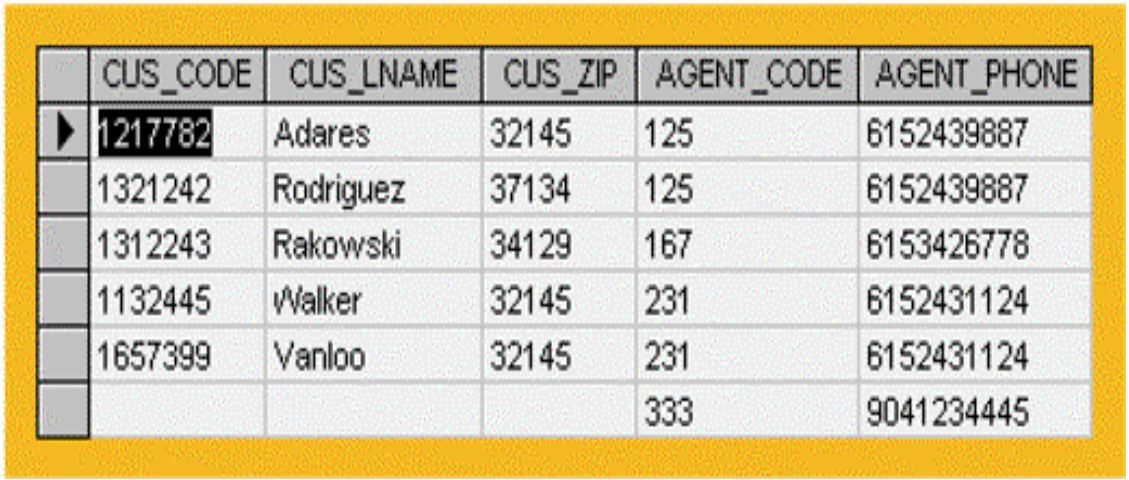
Left Outer Join

FIGURE 3.15 LEFT OUTER JOIN

	CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
▶	1217782	Adares	32145	125	6152439887
	1321242	Rodriguez	37134	125	6152439887
	1312243	Rakowski	34129	167	6153426778
	1132445	Walker	32145	231	6152431124
	1657399	Vanloo	32145	231	6152431124
	1542311	Smithson	37134	421	

Right Outer Join

FIGURE 3.16 RIGHT OUTER JOIN



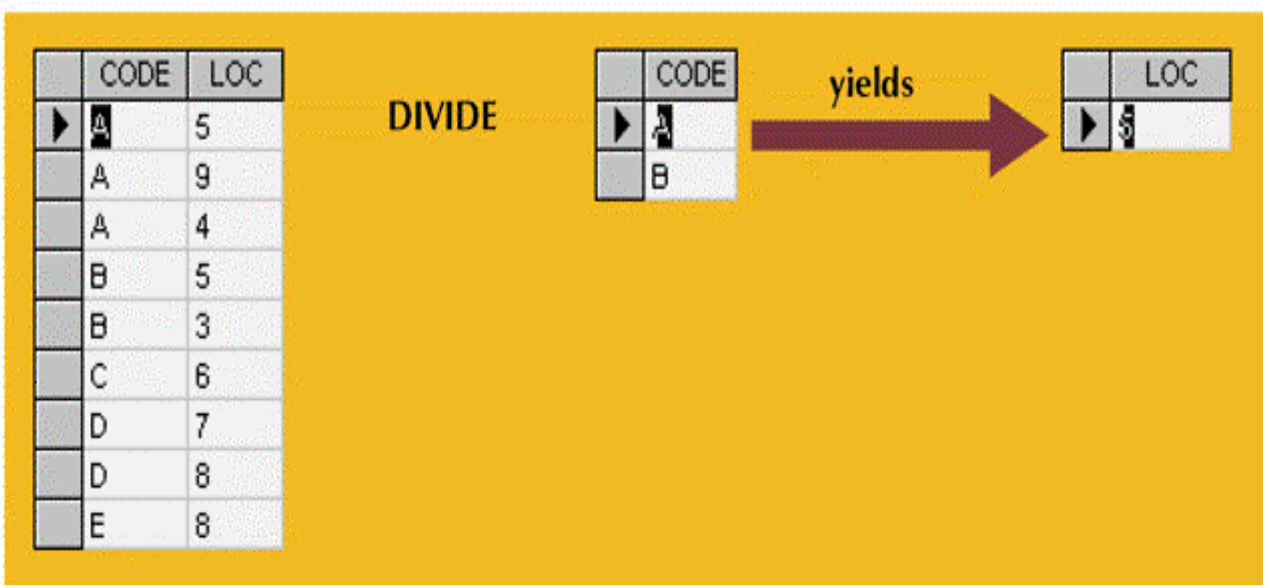
CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124
			333	9041234445

Divide

- DIVIDE requires the use of one single-column table and one two-column table

DIVIDE

FIGURE 3.17 DIVIDE



The Data Dictionary and System Catalog

- Data dictionary
  - Used to provide detailed accounting of all tables found within the user/designer-created database
  - Contains (at least) all the attribute names and characteristics for each table in the system
  - Contains metadata—data about data
  - Sometimes described as “the database designer’s database” because it records the design decisions about tables and their structures

A Sample Data Dictionary

TABLE 3.6 A SAMPLE DATA DICTIONARY

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	REQUIRED	PK Or FK	FK REFERENCED TABLE
CUSTOMER	CUS_CODE	Customer acct. code	CHAR(5)	99999	10000-99999	Y	PK	
	CU_LNAME	Customer last name	VCCHAR(20)	Xxxxxxxx	100-999	Y		
	CUS_FNAME	Customer first name	VCHAR(20	Xxxxxxxx		Y		
	CUS_INITIAL	Customer initial	CHAR(1)	X				
	CUS_RENEW_DATE	Customer insurance renewal date	DATE	dd-mmm-yyyy				
	AGENT_CODE	Agent code	CHAR(3)	999			FK	AGENT
AGENT	AGENT_CODE	Agent code	CHAR(3)	999		Y	PK	
	AGENT_AREACODE	Agent area code	CHAR(3)	999	0.00-9,999,999.99	Y		
	AGENT_PHONE	Agent telephone number	CHAR(8)	999-9999		Y		
	AGENT_LNAME	Agent last name	VCHAR(20)	Xxxxxxxx		Y		
	AGENT_YTD_SALES	Agent year-to-date sales	NUMBER(9,2)	9,999,999.99		Y		

FK = Foreign key

PK = Primary key

CHAR = Fixed character length data (1–255 characters)

VARCHAR = Variable character length data (1–2,000 characters)

NUMBER = Numeric data. NUMBER(9,2) is used to specify numbers with two decimal places and up to nine digits, including the decimal places. Some RDBMSs permit the use of a MONEY or CURRENCY data type.

The Data Dictionary and the System Catalog (continued)

- System catalog
  - Contains metadata
  - Detailed system data dictionary that describes all objects within the database
  - Terms “system catalog” and “data dictionary” are often used interchangeably
  - Can be queried just like any user/designer-created table

Relationships within the Relational Database

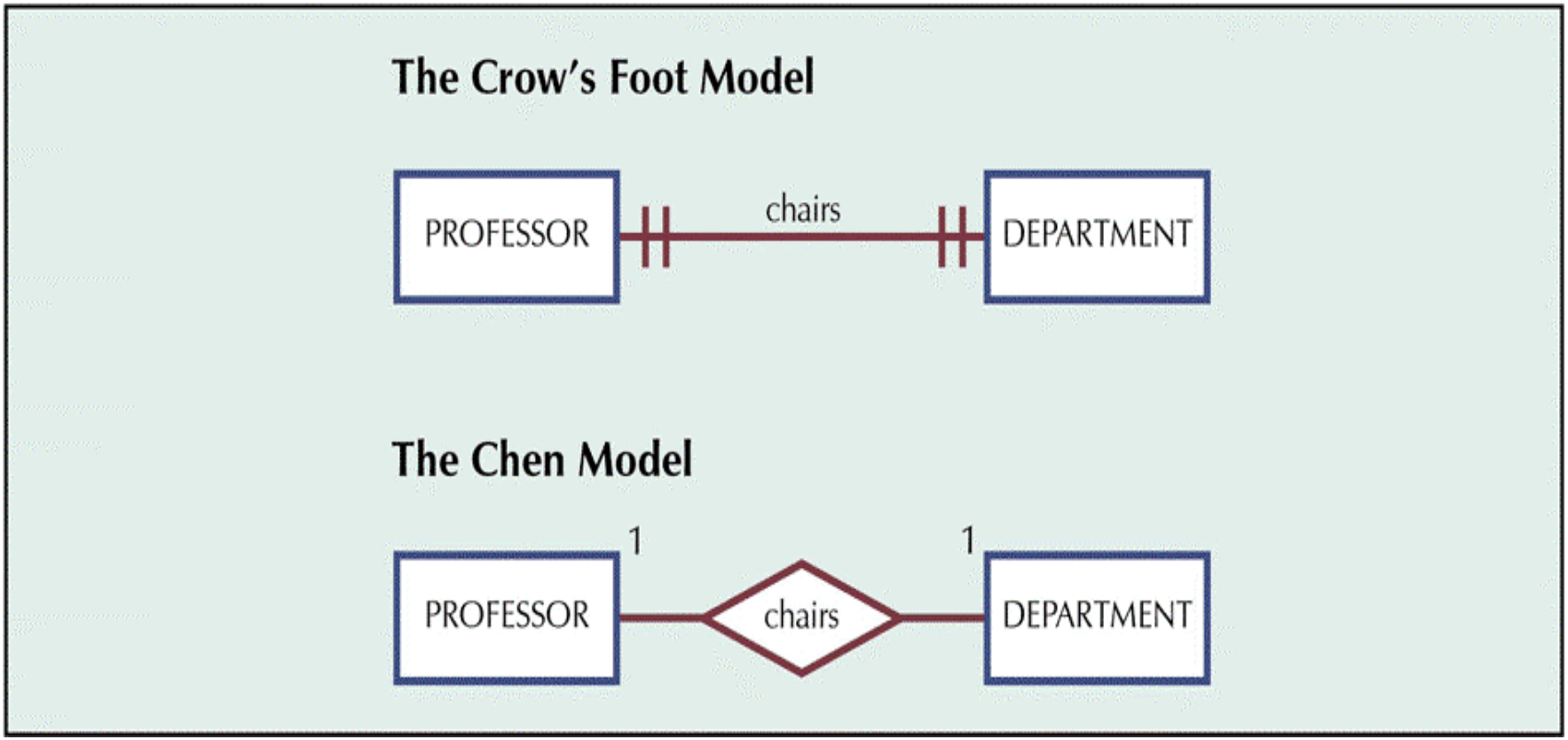
- 1:M relationship
  - Relational modeling ideal
  - Should be the norm in any relational database design
- M:N relationships
  - Must be avoided because they lead to data redundancies
- 1:1 relationship
  - Should be rare in any relational database design

The 1:1 Relationship

- Relational database norm
- Found in any database environment
- One entity can be related to only one other entity, and vice versa
- Often means that entity components were not defined properly
- Could indicate that two entities actually belong in the same table
- Sometimes 1:1 relationships are appropriate

The 1:1 Relationship Between PROFESSOR and DEPARTMENT

FIGURE 3.18 THE 1:1 RELATIONSHIP BETWEEN PROFESSOR AND DEPARTMENT



The Implemented 1:1 Relationship Between PROFESSOR and DEPARTMENT

FIGURE 3.19 THE IMPLEMENTED 1:1 RELATIONSHIP BETWEEN PROFESSOR AND DEPARTMENT

Table name: PROFESSOR **Database name: Ch03_TinyCollege**
Primary key: EMP_NUM
Foreign key: DEPT_CODE

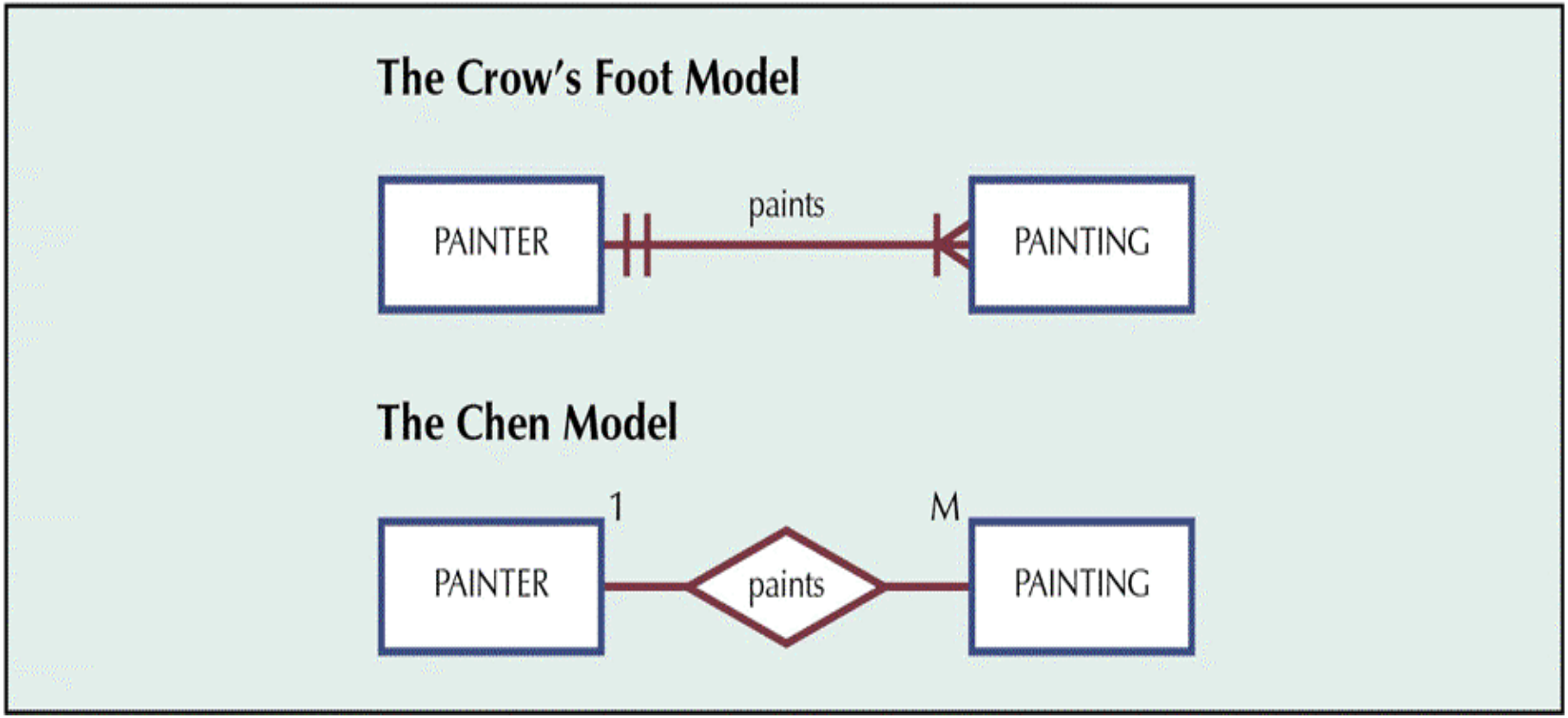
EMP_NUM	DEPT_CODE	PROF_OFFICE	PROF_EXTENSION	PROF_HIGH_DEGREE
103	HIST	DRE 156	6783	Ph.D.
104	ENG	DRE 102	5561	MA
105	ACCT	KLR 229D	8665	Ph.D.
106	MKT/MGT	KLR 126	3899	Ph.D.
110	BIOL	AAK 160	3412	Ph.D.
114	ACCT	KLR 211	4436	Ph.D.
155	MATH	AAK 201	4440	Ph.D.
160	ENG	DRE 102	2248	Ph.D.
162	CIS	KLR 203E	2359	Ph.D.
191	MKT/MGT	KLR 409B	4016	DBA
195	PSYCH	AAK 297	3550	Ph.D.
209	CIS	KLR 333	3421	Ph.D.
228	CIS	KLR 300	3000	Ph.D.
297	MATH	AAK 194	1145	Ph.D.
299	ECON/FIN	KLR 284	2851	Ph.D.
301	ACCT	KLR 244	4683	Ph.D.
335	ENG	DRE 208	2000	Ph.D.
342	SOC	BBG 208	5514	Ph.D.
387	BIOL	AAK 230	8665	Ph.D.
401	HIST	DRE 156	6783	MA
425	ECON/FIN	KLR 284	2851	MBA
435	ART	BBG 185	2278	Ph.D.

Table name: DEPARTMENT
Primary key: DEPT_CODE
Foreign key: EMP_NUM

DEPT_CODE	DEPT_NAME	SCHOOL_CODE	EMP_NUM	DEPT_ADDRESS	DEPT_EXTENSION
ACCT	Accounting	BUS	114	KLR 211, Box 52	3119
ART	Fine Arts	A&SCI	435	BBG 185, Box 128	2278
BIOL	Biology	A&SCI	387	AAK 230, Box 415	4117
CIS	Computer Info. Systems	BUS	209	KLR 333, Box 56	3245
ECON/FIN	Economics/Finance	BUS	299	KLR 284, Box 63	3126
ENG	English	A&SCI	160	DRE 102, Box 223	1004
HIST	History	A&SCI	103	DRE 156, Box 284	1867
MATH	Mathematics	A&SCI	297	AAK 194, Box 422	4234
MKT/MGT	Marketing/Management	BUS	106	KLR 126, Box 55	3342
PSYCH	Psychology	A&SCI	195	AAK 297, Box 438	4110
SOC	Sociology	A&SCI	342	BBG 208, Box 132	2008

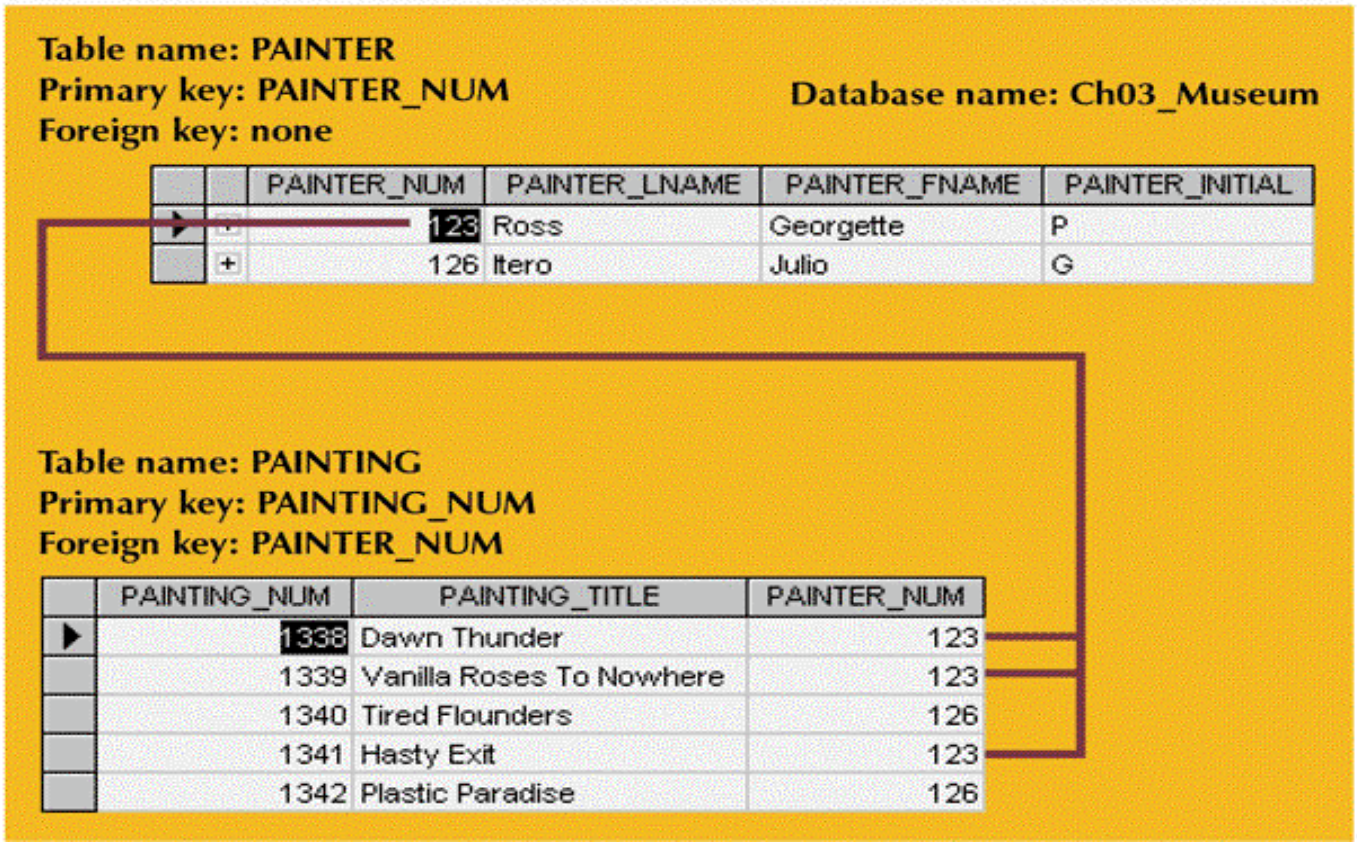
The 1:M Relationship Between PAINTER and PAINTING

FIGURE 3.20 THE 1:M RELATIONSHIP BETWEEN PAINTER AND PAINTING



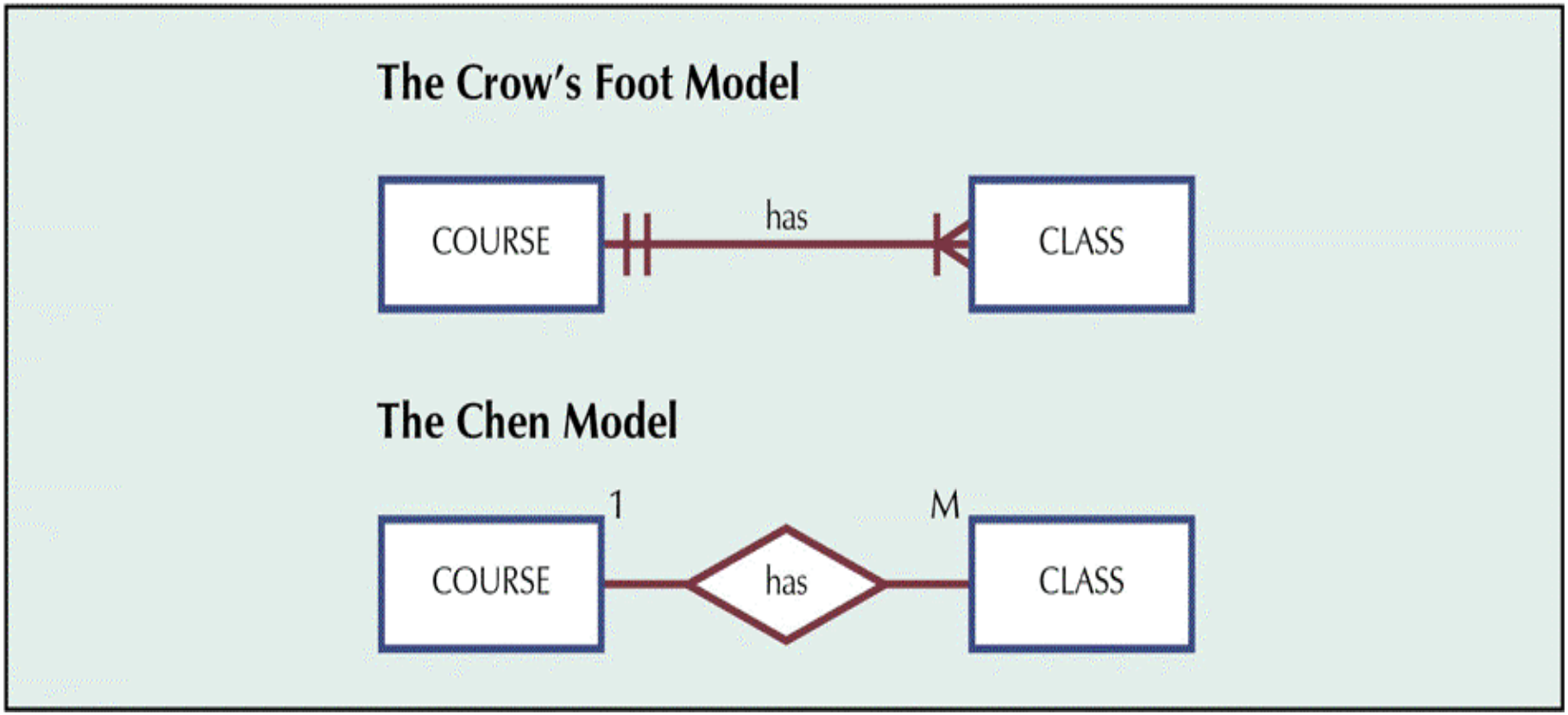
The Implemented 1:M Relationship Between PAINTER and PAINTING

FIGURE 3.21 THE IMPLEMENTED 1:M RELATIONSHIP BETWEEN PAINTER AND PAINTING



The 1:M Relationship Between COURSE and CLASS

FIGURE 3.22 THE 1:M RELATIONSHIP BETWEEN COURSE AND CLASS



The Implemented 1:M Relationship Between COURSE and CLASS

FIGURE 3.23 THE IMPLEMENTED 1:M RELATIONSHIP BETWEEN COURSE AND CLASS

Table name: COURSE
Primary key: CRS_CODE
Foreign key: none
Database name: Ch03_TinyCollege

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Intro. to Microcomputing	3
CIS-420	CIS	Database Design and Implementation	4
QM-261	CIS	Intro. to Statistics	3
QM-362	CIS	Statistical Applications	4

Table name: CLASS
Primary key: CLASS_CODE
Foreign key: CRS_CODE

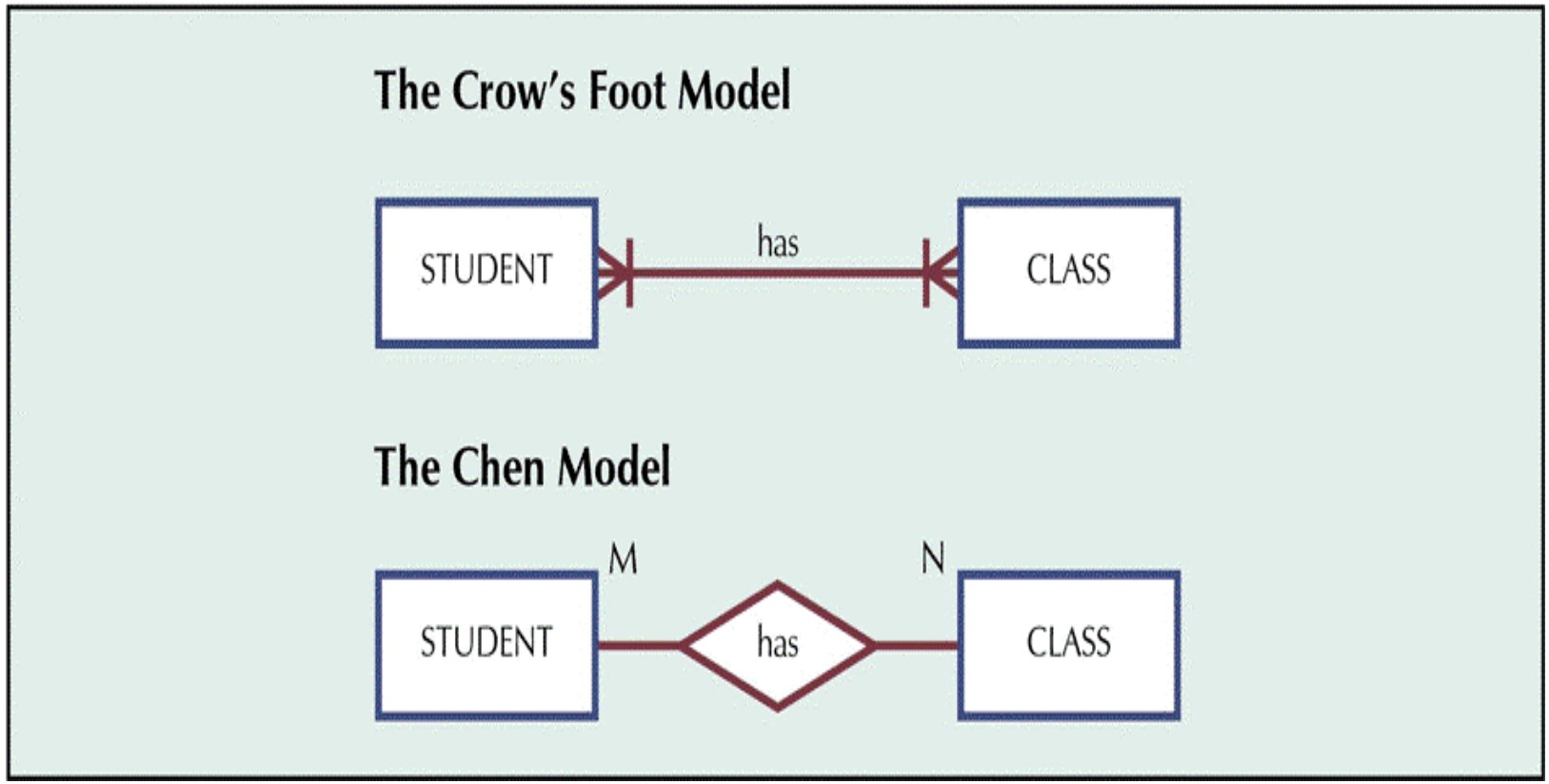
CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10012	ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
10013	ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10015	ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
10016	ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
10017	CIS-220	1	MWF 9:00-9:50 a.m.	KLR209	228
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10019	CIS-220	3	MWF 10:00-10:50 a.m.	KLR209	228
10020	CIS-420	1	W 6:00-8:40 p.m.	KLR209	162
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10022	QM-261	2	TTh 1:00-2:15 p.m.	KLR200	114
10023	QM-362	1	MWF 11:00-11:50 a.m.	KLR200	162
10024	QM-362	2	TTh 2:30-3:45 p.m.	KLR200	162

The M:N Relationship

- Can be implemented by breaking it up to produce a set of 1:M relationships
- Can avoid problems inherent to M:N relationship by creating a composite entity or bridge entity

The ERD's M:N Relationship Between STUDENT and CLASS

FIGURE 3.24 THE ERD'S M:N RELATIONSHIP BETWEEN STUDENT AND CLASS



Sample Student Enrollment Data

TABLE 3.7 SAMPLE STUDENT ENROLLMENT DATA

STUDENT'S LAST NAME	SELECTED CLASSES
Bowser	Accounting 1, ACCT-211, code 10014 Intro. to Microcomputing, CIS-220, code 10018 Intro. To Statistics, QM-261, code 10021
Smithson	Accounting 1, ACCT-211, code 10014 Intro. to Microcomputing, CIS-220, code 10018 Intro. To Statistics, QM-261, code 10021

The M:N Relationship Between STUDENT and CLASS

FIGURE 3.25 THE M:N RELATIONSHIP BETWEEN STUDENT AND CLASS

Table name: STUDENT
Primary key: STU_NUM
Foreign key: none
Database name: Ch03_CollegeTry

STU_NUM	STU_LNAME	CLASS_CODE
321452	Bowser	10014
321452	Bowser	10018
321452	Bowser	10021
324257	Smithson	10014
324257	Smithson	10018
324257	Smithson	10021

Table name: CLASS
Primary key: CLASS_CODE
Foreign key: STU_NUM

CLASS_CODE	STU_NUM	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	321452	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10014	324257	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	321452	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10018	324257	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	321452	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10021	324257	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

Linking Table

- Implementation of a composite entity
- Yields required M:N to 1:M conversion
- Composite entity table must contain at least the primary keys of original tables
- Linking table contains multiple occurrences of the foreign key values
- Additional attributes may be assigned as needed

Converting the M:N Relationship into Two 1:M Relationships

FIGURE 3.26 CONVERTING THE M:N RELATIONSHIP INTO TWO 1:M RELATIONSHIPS

Table name: STUDENT
Primary key: STU_NUM
Foreign key: none

Database name: Ch03_CollegeTry2

STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

Table name: ENROLL
Primary key: CLASS_CODE + STU_NUM
Foreign key: CLASS_CODE, STU_NUM

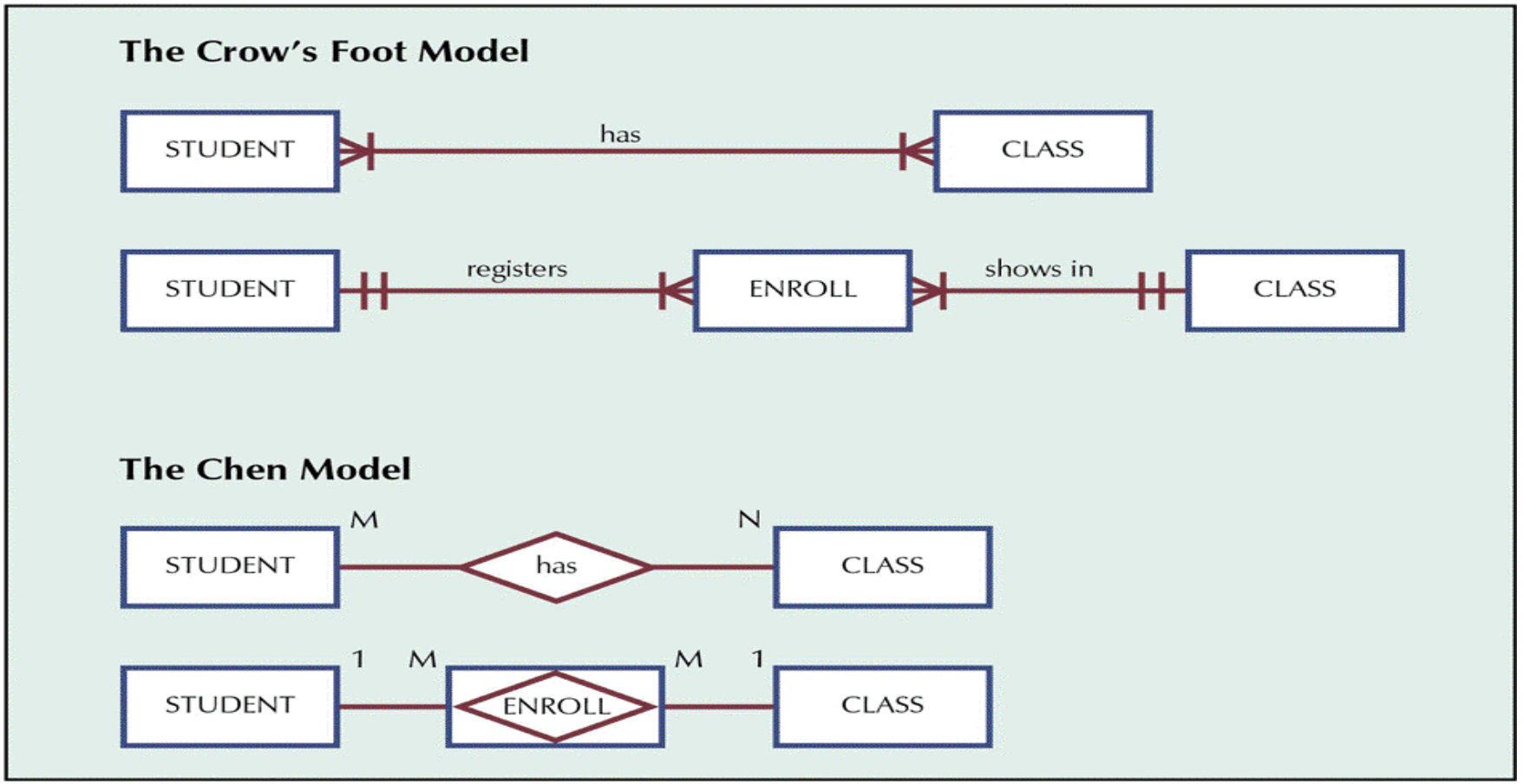
CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

Table name: CLASS
Primary key: CLASS_CODE
Foreign key: CRS_CODE

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

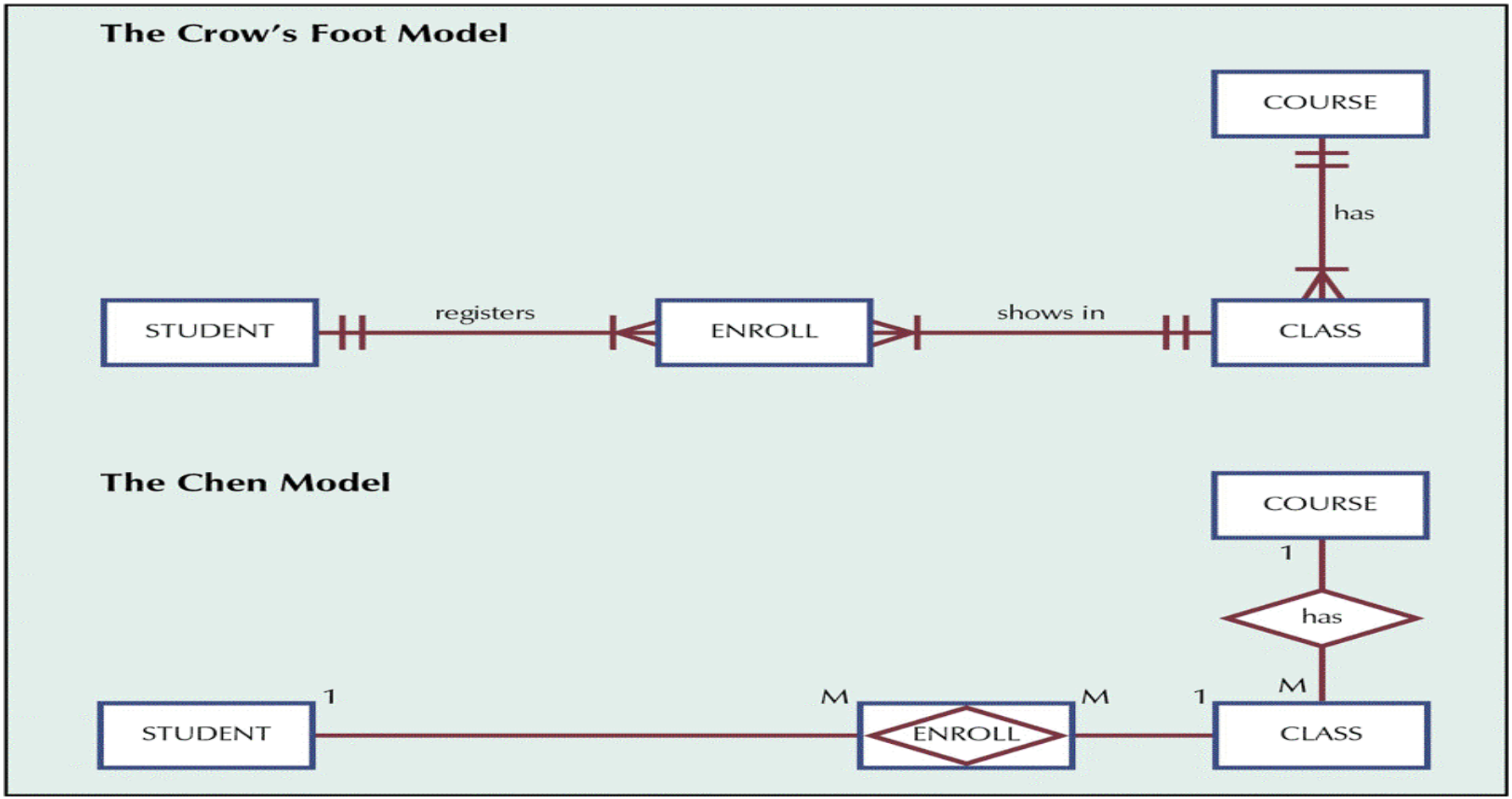
Changing the M:N Relationship to Two 1:M Relationships

FIGURE 3.27 CHANGING THE M:N RELATIONSHIP TO TWO 1:M RELATIONSHIPS



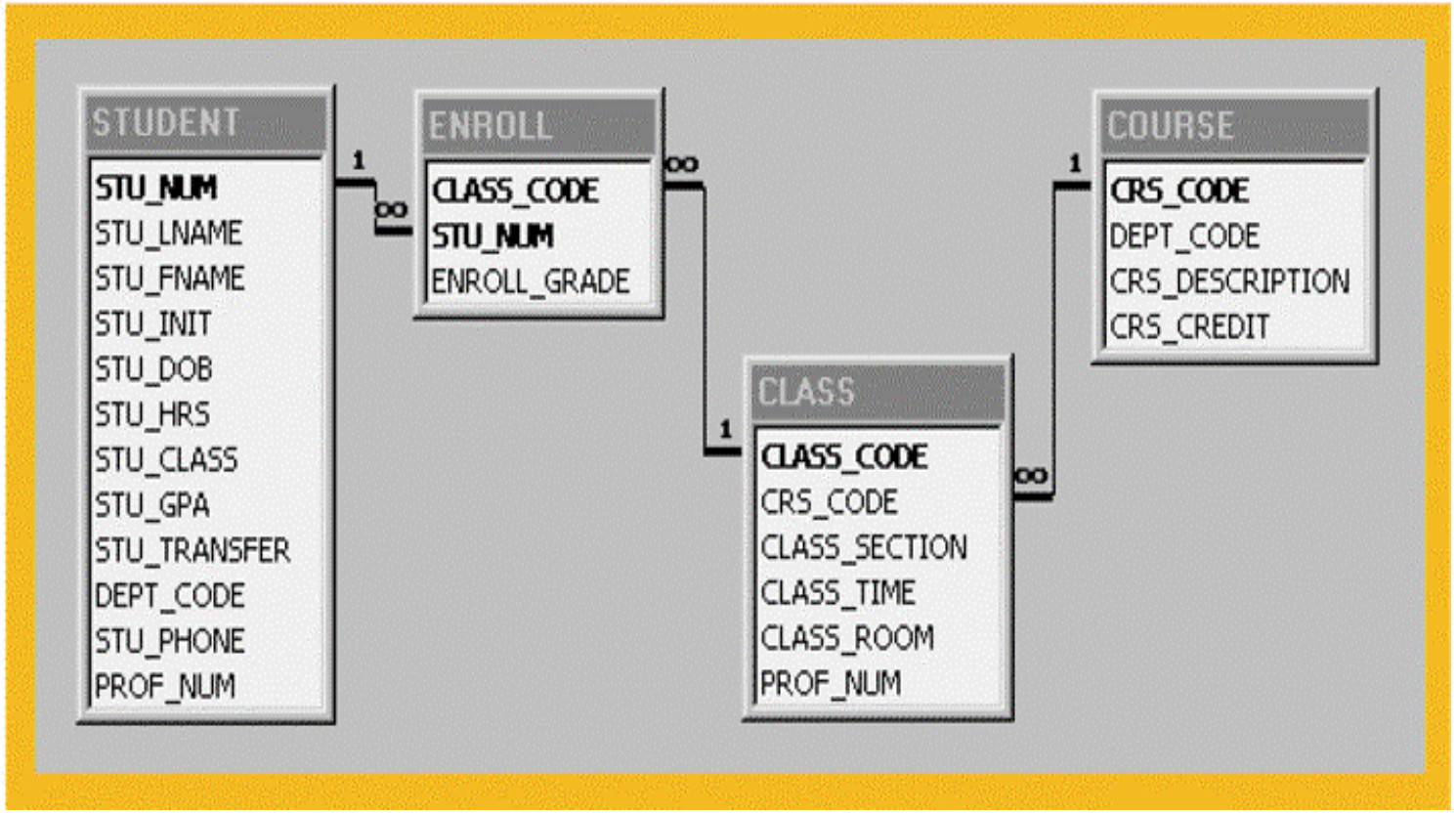
The Expanded Entity Relationship Model

FIGURE 3.28 THE EXPANDED ENTITY RELATIONSHIP MODEL



The Relational Schema for the Ch03_TinyCollege Database

FIGURE 3.29 THE RELATIONAL SCHEMA FOR THE CH03_TINYCOLLEGE DATABASE



Data Redundancy Revisited

- Data redundancy leads to data anomalies
  - Such anomalies can destroy database effectiveness
- Foreign keys
  - Control data redundancies by using common attributes shared by tables
  - Crucial to exercising data redundancy control
- Sometimes, data redundancy is necessary

A Small Invoicing System

FIGURE 3.30 A SMALL INVOICING SYSTEM

Table name: CUSTOMER
Primary key: CUS_CODE
Foreign key: none
Database name: Ch03_SaleCo

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE
10010	Ramas	Alfred	A	615	844-2573
10011	Dunne	Leona	K	713	894-1238
10012	Smith	Kathy	W	615	894-2285
10013	Olowski	Paul	F	615	894-2180
10014	Orlando	Myron		615	222-1672
10015	O'Brian	Amy	B	713	442-3381
10016	Brown	James	G	615	297-1228
10017	Williams	George		615	290-2556
10018	Farriss	Anne	G	713	382-7185
10019	Smith	Olette	K	615	297-3809

Table name: INVOICE
Primary key: INV_NUMBER
Foreign key: CUS_CODE

INV_NUMBER	CUS_CODE	INV_DATE
1001	10014	08-Mar-04
1002	10011	08-Mar-04
1003	10012	08-Mar-04
1004	10011	09-Mar-04

Table name: LINE
Primary key: INV_NUMBER + LINE_NUMBER
Foreign keys: INV_NUMBER, PROD_CODE

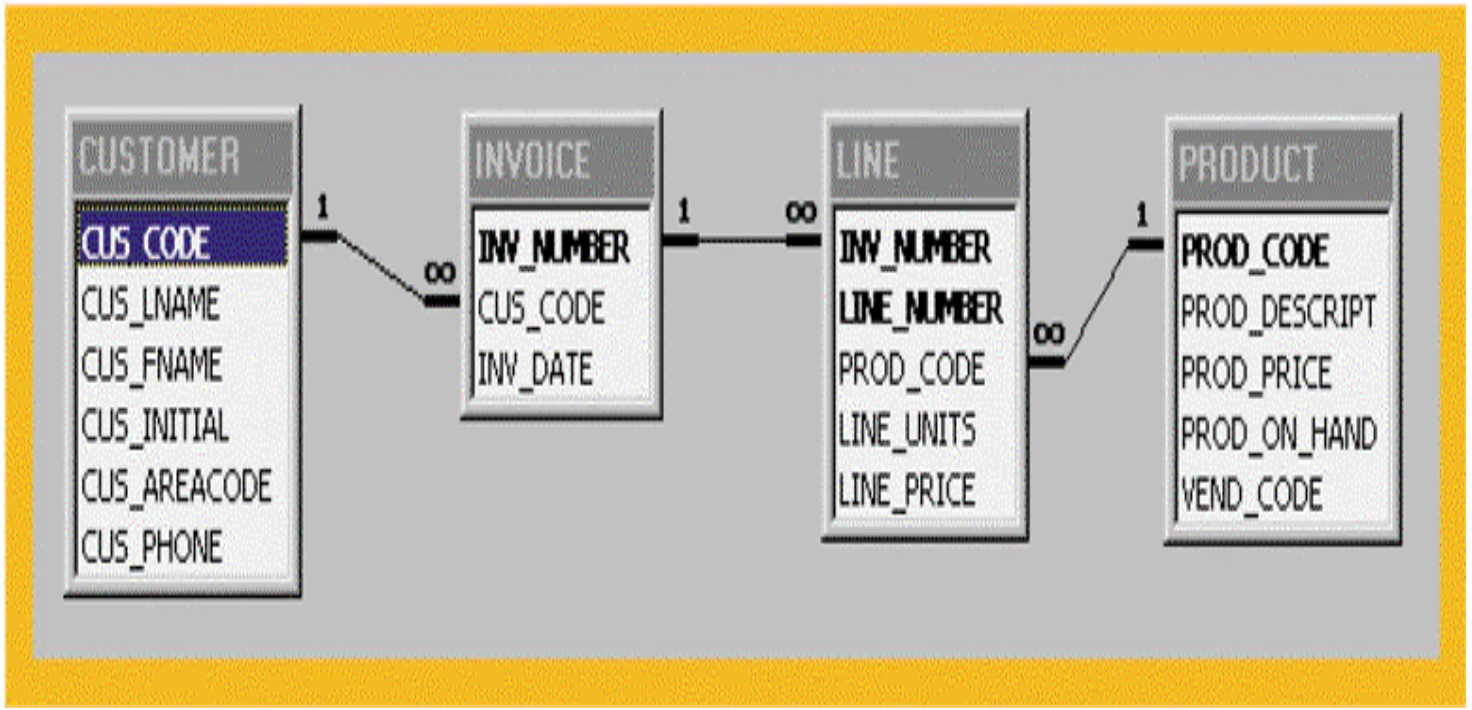
INV_NUMBER	LINE_NUMBER	PROD_CODE	LINE_UNITS	LINE_PRICE
1001	1	123-21UUY	1	$189.99
1001	2	SRE-657UG	3	$2.99
1002	1	QER-34256	2	$18.63
1003	1	ZZX/3245Q	1	$6.79
1003	2	SRE-657UG	1	$2.99
1003	3	001278-AB	1	$12.95
1004	1	001278-AB	1	$12.95
1004	2	SRE-657UG	2	$2.99

Table name: PRODUCT
Primary key: PROD_CODE
Foreign key: none

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

The Relational Schema for the Invoicing System

FIGURE 3.31 THE RELATIONAL SCHEMA FOR THE INVOICING SYSTEM

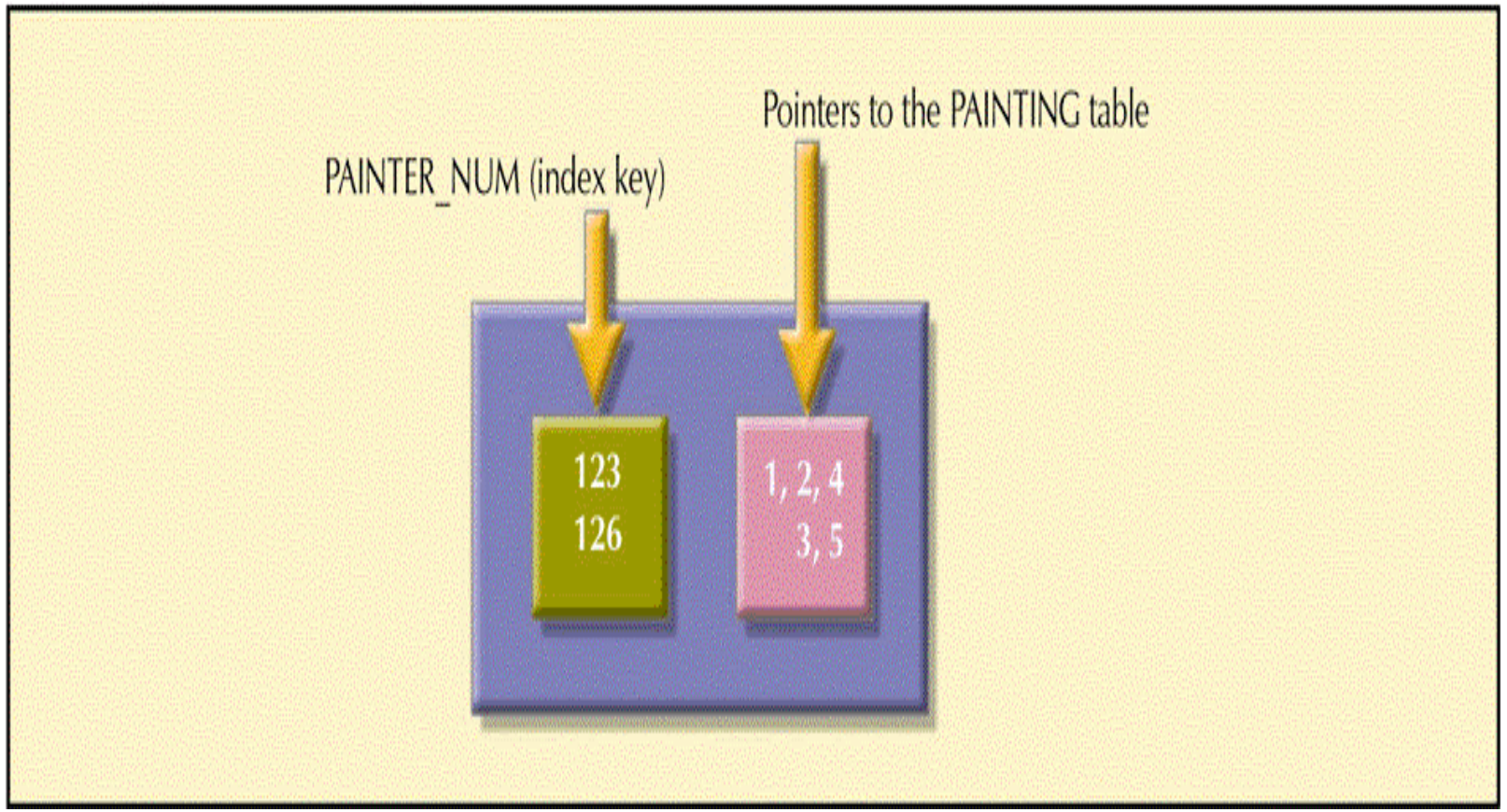


Indexes

- Arrangement used to logically access rows in a table
- Index key
  - Index's reference point
  - Points to data location identified by the key
- Unique index
  - Index in which the index key can only have one pointer value (row) associated with it
- Each index is associated with only one table

Components of an Index

FIGURE 3.32 COMPONENTS OF AN INDEX



Summary

- Entities are basic building blocks of a relational database
- Entity set is a grouping of related entities, stored in a table
- Keys define functional dependencies
  - Superkey
  - Candidate key
  - Primary key
  - Secondary key
  - Foreign key

Summary (continued)

- Primary key uniquely identifies attributes
  - Can link tables by using controlled redundancy
- Relational databases classified according to degree to which they support relational algebra functions
- Relationships between entities are represented by entity relationship models
- Data retrieval speed can be increased dramatically by using indexes