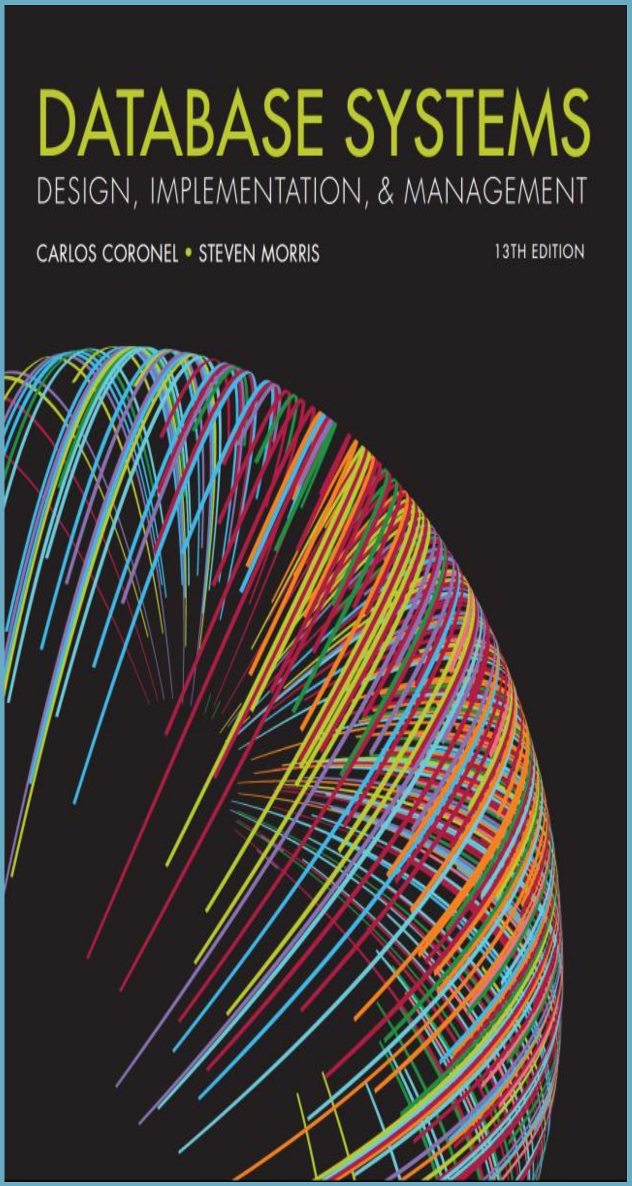

DATABASE SYSTEMS
DESIGN, IMPLEMENTATION, & MANAGEMENT
CARLOS CORONEL • STEVEN MORRIS
13TH EDITION

Chapter 3

The Relational Database Model

The Relational Data Model

OBJECTIVES

- Describe relations and tables
- Describe the composition and features of a table
- Understand and apply the concept of keys
- Describe the use of relational set operators
- Describe and create a data dictionary
- Explain relationships
- Remove M:N relationships

The Relational Data Model

Allows us to logically describe how we will view data

Think of a relation as a table containing a group of related entity occurrences.

Relation = Entity set = Table (approximately)

Table consists of rows and columns

STU_ID	STU_LNAME	STU_FNAME	STU_DOB	STU_STATUS	STU_GPA
83675	Bloggs	Joe	1990-04-04	Normal	3.20
93467	Smith	Gillian	1989-12-29	Honours	3.97
94324	Johnson	Andy	1992-09-11	Normal	2.55
96223	Johnson	Andy	1985-06-14	Probation	1.80

Characteristics of a Relational Table

- Two Dimensional Structure composed of rows and columns
- Each row (*tuple*) represents a single entity
- Each column represents an attribute and has a distinct name
- The intersection of row and column represents a single data value
- Values for a single column must have the same data format
- Each column has a range of values (*attribute domain*)
- The order of the rows and columns don't matter to the DBMS
- Each table must have an attribute or attributes that uniquely identify a row

0.00 – 4.00

STU_ID	STU_LNAME	STU_FNAME	STU_DOB	STU_STATUS	STU_GPA
83675	Bloggs	Joe	1990-04-04	Normal	3.20
93467	Smith	Gillian	1989-12-29	Honours	3.97
94324	Johnson	Andy	1992-09-11	Normal	2.55
96223	Johnson	Andy	1985-06-14	Probation	1.80

Keys & Dependencies

- Key is one or more attributes that uniquely identify a row
- Functional Dependence means the value of one or more attributes determines the value of one or more other attributes

key

STU_ID -> (STU_LNAME, STU_FNAME)
(STU_ID, STU_DOB) -> STU_STATUS

key

STU_ID	STU_LNAME	STU_FNAME	STU_DOB	STU_STATUS	STU_GPA
83675	Bloggs	Joe	1990-04-04	Normal	3.20
93467	Smith	Gillian	1989-12-29	Honours	3.97
94324	Johnson	Andy	1992-09-11	Normal	2.55
96223	Johnson	Andy	1985-06-14	Probation	1.80

Types of Keys

Composite Key

- more than one attribute
- attributes that are part of a composite are *key attributes*

Super key

- any key that can uniquely identify any row in the table

Candidate key

- a minimal super key - no extra attributes

Primary key

- Must be unique
- must not be null

Secondary key

- used for data retrieval
- not necessarily unique

Foreign key

- must match the primary key in another table (or be null)

Types of Keys

Composite
Super key
Candidate key
Primary key
Secondary key
Foreign key

STU_ID	STU_LNAME	STU_FNAME	STU_DOB	STU_STATUS	STU_GPA
83675	Bloggs	Joe	1990-04-04	Normal	3.20
93467	Smith	Gillian	1989-12-29	Honours	3.97
94324	Johnson	Andy	1992-09-11	Normal	2.55
96223	Johnson	Andy	1985-06-14	Probation	1.80

Composite key

- more than one attribute
- attributes that are part of a composite are *key attributes*

Types of Keys

Composite
Super key
Candidate key
Primary key
Secondary key
Foreign key

STU_ID	STU_LNAME	STU_FNAME	STU_DOB	STU_STATUS	STU_GPA
83675	Bloggs	Joe	1990-04-04	Normal	3.20
93467	Smith	Gillian	1989-12-29	Honours	3.97
94324	Johnson	Andy	1992-09-11	Normal	2.55
96223	Johnson	Andy	1985-06-14	Probation	1.80

Super key

Any key that can uniquely identify any row in the table

Types of Keys

Composite
Super key
Candidate key
Primary key
Secondary key
Foreign key

STU_ID	STU_LNAME	STU_FNAME	STU_DOB	STU_STATUS	STU_GPA
83675	Bloggs	Joe	1990-04-04	Normal	3.20
93467	Smith	Gillian	1989-12-29	Honours	3.97
94324	Johnson	Andy	1992-09-11	Normal	2.55
96223	Johnson	Andy	1985-06-14	Probation	1.80

Candidate key

A minimal super key - no extra attributes

Types of Keys

Composite
Super key
Candidate key
Primary key
Secondary key
Foreign key

STU_ID	STU_LNAME	STU_FNAME	STU_DOB	STU_STATUS	STU_GPA
83675	Bloggs	Joe	1990-04-04	Normal	3.20
93467	Smith	Gillian	1989-12-29	Honours	3.97
94324	Johnson	Andy	1992-09-11	Normal	2.55
96223	Johnson	Andy	1985-06-14	Probation	1.80

Primary key
Must be unique
must not be null

UNIQUE
No part is NULL
is a candidate key
You pick.

Types of Keys

Composite
Super key
Candidate key
Primary key
Secondary key
Foreign key

STU_ID	STU_LNAME	STU_FNAME	STU_DOB	STU_STATUS	STU_GPA
83675	Bloggs	Joe	1990-04-04	Normal	3.20
93467	Smith	Gillian	1989-12-29	Honours	3.97
94324	Johnson	Andy	1992-09-11	Normal	2.55
96223	Johnson	Andy	1985-06-14	Probation	1.80

Secondary

Secondary key

- used for data retrieval
- not necessarily unique

Types of Keys

Composite
Super key
Candidate key
Primary key
Secondary key
Foreign key

PK

STU_ID	STU_LNAME	STU_FNAME	STU_DOB	STU_STATUS	STU_GPA
83675	Bloggs	Joe	1990-04-04	Normal	3.20
93467	Smith	Gillian	1989-12-29	Honours	3.97
94324	Johnson	Andy	1992-09-11	Normal	2.55
96223	Johnson	Andy	1985-06-14	Probation	1.80

PK

FK

Reg ID	CRS_ID	CRS_NAME	STU_ID
1	CIT163-13WN-C01	Database	83675
2	CIT163-13WN-C02	Database	93467
3	CIT163-13WN-C01	Database	96332
4	CIT 163-10WNC02	Database	NULL

Foreign key
must match the primary key in another table (or be null)

Integrity Rules

Entity Integrity — Primary keys

All primary key entries are unique
No part of a primary key may be null

Referential Integrity — Foreign keys

matches a primary key value in a related table
may be null if it doesn't violate entity integrity

Relational Set Operators

All relational operators have closure → The result is another relation

Select
Project
Union
Intersect
Difference
Product
Join
Divide

Select

"Unary" - uses a single table as input
Yields values all of the rows meeting criteria
- Includes ALL attributes (columns)
- "Horizontal" selection

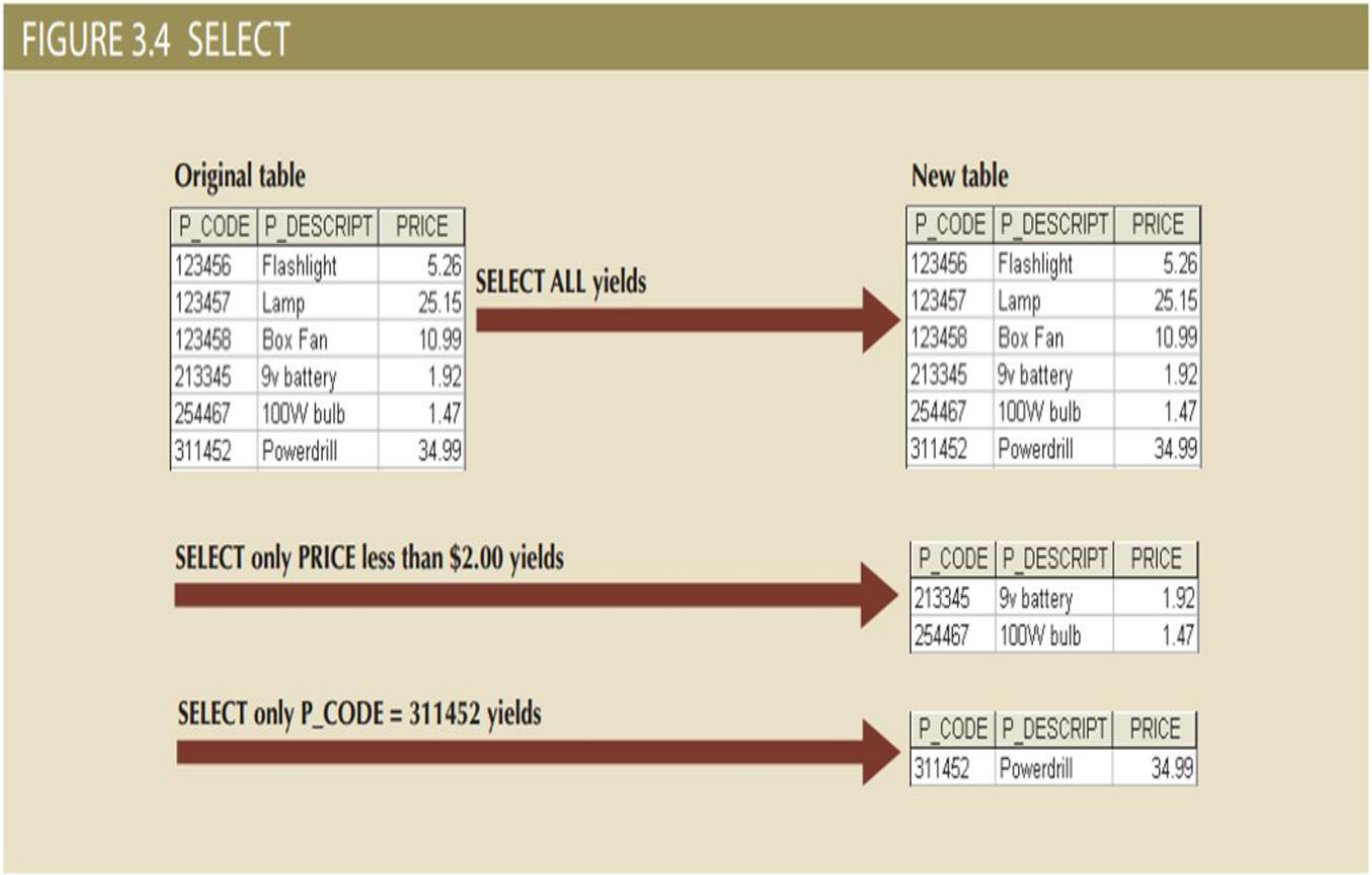


FIGURE 3.4 SELECT

Original table

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

SELECT ALL yields

New table

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

SELECT only PRICE less than $2.00 yields

P_CODE	P_DESCRIPT	PRICE
213345	9v battery	1.92
254467	100W bulb	1.47

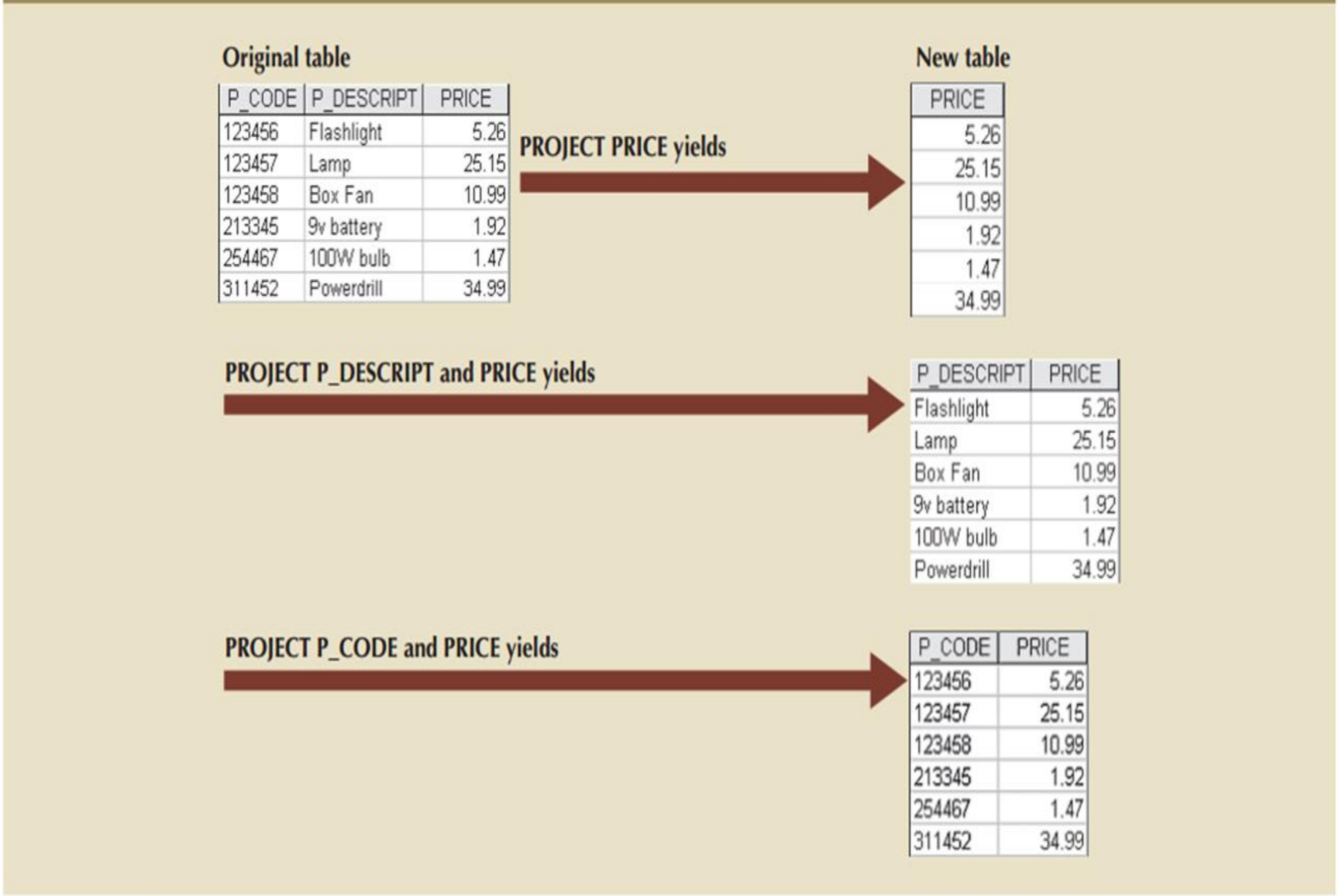
SELECT only P_CODE = 311452 yields

P_CODE	P_DESCRIPT	PRICE
311452	Powerdrill	34.99

Project

"Unary" - uses a single table as input
Yields values all of the columns meeting criteria
- Includes ALL rows
- "Vertical" selection

FIGURE 3.5 PROJECT



Original table

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

PROJECT PRICE yields

New table

PRICE
5.26
25.15
10.99
1.92
1.47
34.99

PROJECT P_DESCRIPT and PRICE yields

P_DESCRIPT	PRICE
Flashlight	5.26
Lamp	25.15
Box Fan	10.99
9v battery	1.92
100W bulb	1.47
Powerdrill	34.99

PROJECT P_CODE and PRICE yields

P_CODE	PRICE
123456	5.26
123457	25.15
123458	10.99
213345	1.92
254467	1.47
311452	34.99

Union

merges two relations
with the same attributes
eliminates duplicates

$$\begin{array}{l} \{A\ B\ C\ D\} \\ \cup\ \{A\ \ C\ \ E\} \\ \hline \{A\,B\,C\,D\,E\} \end{array}$$

X

name	Phone
A	123
B	456

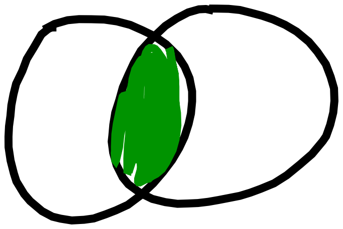
∪

Y

name	Phone
C	456
B	789
A	123

name	Phone
A	123
B	456
B	789
C	456

Intersect



new relation where entities are in **both**

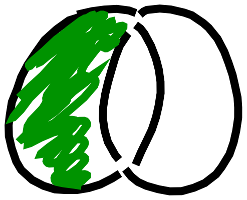
name	Phone
A	123
B	456

$\cap$

name	Phone
A	123
C	456
D	789

name	Phone
A	123

Difference



In the first relation but NOT in the second

name	Phone
~~A~~	~~123~~
B	456
C	789

—

name	Phone
A	123
C	456
D	789

Name	Phone
B	456
C	789

Product

- "Multiplies" two relations

name	phone
A	123
B	456
C	789

×

name	address
A	Aspen
B	Birch

Name_1	Phone	name_2	address
A	123	A	Aspen
A	123	B	Birch
B	456	A	Aspen
B	456	B	Birch
C	789	A	Aspen
C	789	B	Birch

Join

- Allows information from 2 or more independent tables to be combined
- They are linked by common attributes
- Yields a table that does NOT contain unmatched pairs, only the copies of the matches

Product + Select + Project

name	phone
A	123
B	456
C	789

"NAural" join

name	address
A	Aspen
B	Birch
D	Dogwood

Name	phone	address
A	123	ASPEN
B	456	Birch

Types of Joins

inner → matches both sides

Outer → All of one side and only matches from the other

Inner Join

on s.sid = g.sid

S

pk

sid	name	age
1	Joe	11
2	Sally	22
3	Jim	28

G

FK pk

sid	groupId	groupName
1	A	Alphas
3	B	Betas
	G	Gammas

s.sid	s.name	s.age	g.sid	g.groupId	g.groupName
1	Joe	11	1	A	Alphas
3	Jim	28	3	B	Betas

Left Outer Join

on s.sid = g.sid

S

PK

sid	name	age
1	Joe	11
2	Sally	22
3	Jim	28

G

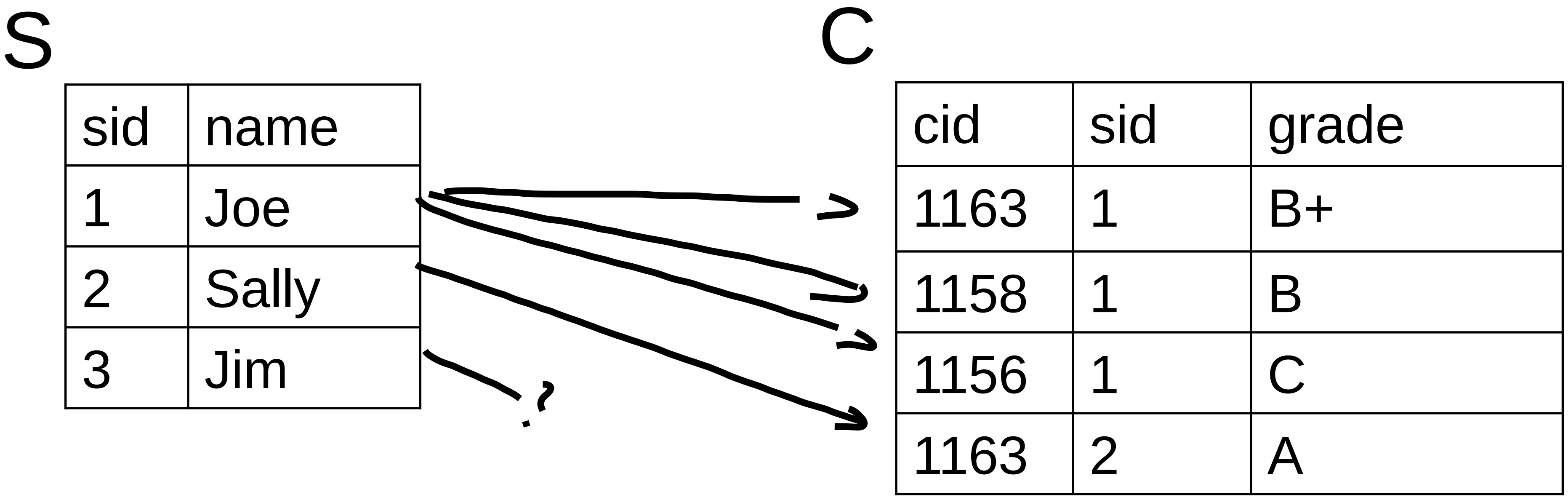
FK PK

sid	groupId	groupName
1	A	Alphas
3	B	Betas
4	G	Gammas

s.sid	s.name	s.age	g.sid	g.groupId	g.groupName
1	Joe	11	1	A	Alphas
2	Sally	22			
3	Jim	28	3	B	Betas

Left Outer Join

on s.sid = c.sid



S

sid	name
1	Joe
2	Sally
3	Jim

C

cid	sid	grade
1163	1	B+
1158	1	B
1156	1	C
1163	2	A

s.sid	s.name	c.cid	c.sid	c.grade
1	Joe	1163	1	B+
1	Joe	1158	1	B
1	Joe	1156	1	C
2	Sally	1163	2	A
3	Jim			

Right Outer Join

on s.sid = g.sid

S

sid	name	age
1	Joe	11
2	Sally	22
3	Jim	28

?

G

sid	groupId	groupName
1	A	Alphas
2	B	Betas
	G	Gammas

s.sid	s.name	s.age	g.sid	g.groupId	g.groupName
1	Joe	11	1	A	Alphas
2	Sally	22	2	B	Betas
				G	GAMMAS

name	phone
A	123
B	456
C	789

left outer join

left join

name	address
A	Aspen
B	Birch
D	Dogwood

name	Phone	address
A	123	Aspen
B	456	Birch
C	789	

name	phone
A	123
B	456
C	789

right outer join

right join

name	address
A	Aspen
B	Birch
D	Dogwood

Name	address	Phone
A	Aspen	123
B	Birch	456
D	Dogwood	

name	phone
A	123
B	456
C	789

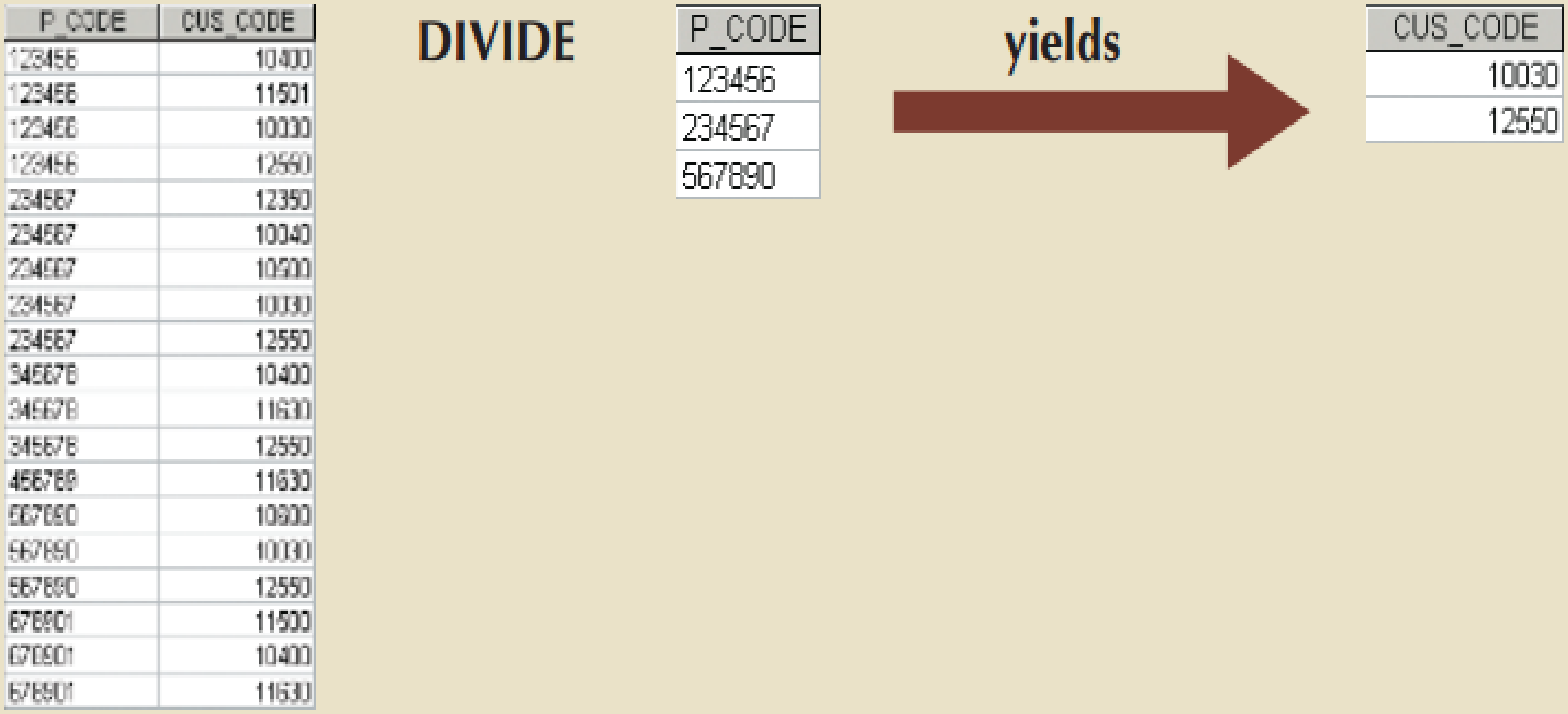
full outer join

name	address
A	Aspen
B	Birch
D	Dogwood

name	Phone	address
A	123	Aspen
B	456	Birch
C	789	
D		Dogwood

- Divide
  - Uses one double-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

GURE 3.16 DIVIDE



P_CODE	CUS_CODE
123456	10400
123456	11501
123456	10030
123456	12550
234567	12350
234567	10040
234567	10900
234567	10030
234567	12550
345678	10400
345678	11630
345678	12550
456789	11630
567890	10900
567890	10030
567890	12550
678901	11500
678901	10400
678901	11630

DIVIDE

P_CODE
123456
234567
567890

yields

CUS_CODE
10030
12550

Data Dictionary

The designers description

Look out for synonyms and homonyms

Homonym: same name is used to label different attributes

Synonym: different names are used to describe the same attribute

STU_ID	STU_LNAME	STU_FNAME	STU_DOB	STU_STATUS	STU_GPA
83675	Bloggs	Joe	1990-04-04	Normal	3.20
93467	Smith	Gillian	1989-12-29	Honours	3.97
94324	Johnson	Andy	1992-09-11	Normal	2.55
96223	Johnson	Andy	1985-06-14	Probation	1.80

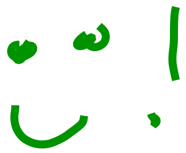
CRS_ID	CRS_NAME	STU_ID
CIT163-13WN-C01	Database	83675
CIT163-13WN-C02	Database	93467
CIT163-13WN-C01	Database	96332

Table Name	Attribute name	Contents (descrip)	Type	Format	Range	REQ	PK/FK	FK Ref

Relationships

1:M

This is the ideal!



1:1

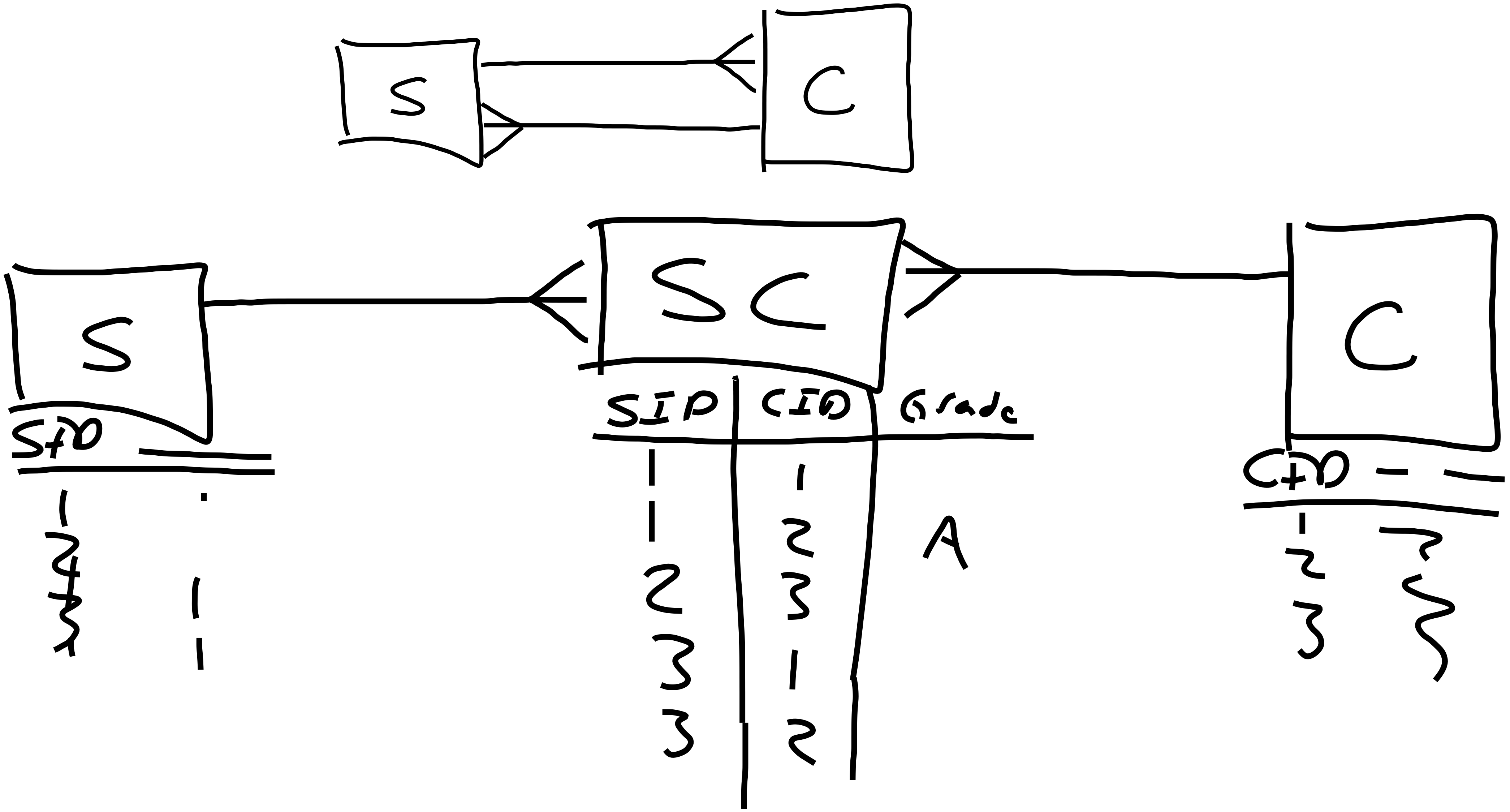
Should be rare

M:N

cannot be implemented

How to convert M:N into 1:M

More FKs do not fix the problem
Create a composite entity (associative, bridge)



SID
1
2
3

SID	CID	Grade
1	1	
1	2	A
2	3	
3	1	
3	2	

CID
1
2
3

FIGURE 3.26 CHANGING THE M:N RELATIONSHIPS TO TWO 1:M RELATIONSHIPS

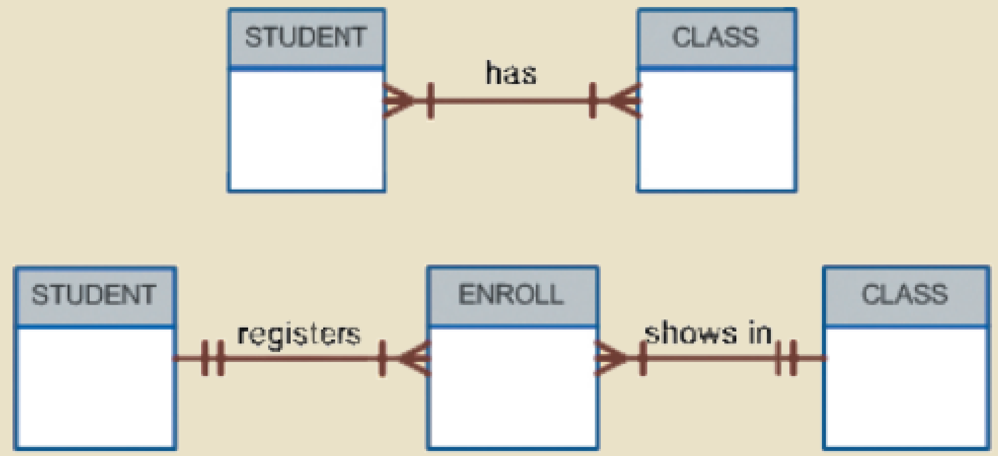
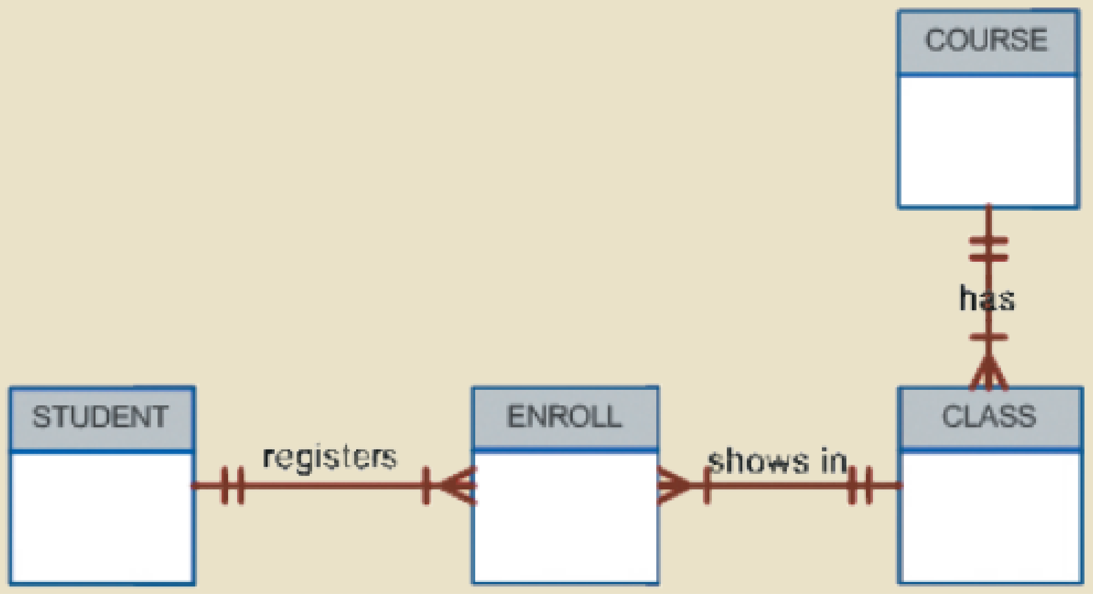


FIGURE 3.27 THE EXPANDED ER MODEL



A note about data redundancy

Foreign keys appear redundant but actually reduce overall redundancy

Codd’s Relational Database Rules (1 of 2)

Table 13.8	**Dr. Codd’s 12 Relational Database Rules**	
Rule	Rule Name	Description
1	Information	All information in a relational database must be logically represented as column values in rows within tables.
2	Guaranteed access	Every value in a table is guaranteed to be accessible through a combination of table name, primary key value, and column name.
3	Systematic treatment of nulls	Nulls must be represented and treated in a systematic way, independent of data type.
4	Dynamic online catalog based on the relational model	The metadata must be stored and managed as ordinary data—that is, in tables within the database; such data must be available to authorized users using the standard database relational language.
5	Comprehensive data sublanguage	The relational database may support many languages; however, it must support one well-defined, declarative language as well as data definition, view definition, data manipulation (interactive and by program), integrity constraints, authorization, and transaction management (begin, commit, and rollback).
6	View updating	Any view that is theoretically updatable must be updatable through the system.
7	High-level insert, update, and delete	The database must support set-level inserts, updates, and deletes.

Codd’s Relational Database Rules (2 of 2)

Table 13.8	**Dr. Codd’s 12 Relational Database Rules**	
Rule	Rule Name	Description
8	Physical data independence	Application programs and ad hoc facilities are logically unaffected when physical access methods or storage structures are changed.
9	Logical data independence	Application programs and ad hoc facilities are logically unaffected when changes are made to the table structures that preserve the original table values (changing order of columns or inserting columns).
10	Integrity independence	All relational integrity constraints must be definable in the relational language and stored in the system catalog, not at the application level.
11	Distribution independence	The end users and application programs are unaware of and unaffected by the data location (distributed vs. local databases).
12	Nonsubversion	If the system supports low-level access to the data, users must not be allowed to bypass the integrity rules of the database.
13	Rule zero	All preceding rules are based on the notion that to be considered relational, a database must use its relational facilities exclusively for management.