LECTURE 04

Relational and Relational Algebra

What is Relational Algebra?

- It is a language in which we can ask questions (query) of a database.
- Basic premise is that *tables are sets* (mathematical) and so our query language should manipulate sets with ease.
- Traditional Set Operations:
  - union, intersection, Cartesian product, set difference.
- Extended Set Operations:
  - selection, projection, join, quotient.

Relational Database Operators

- **Relational algebra**
  - Defines theoretical way of manipulating table contents using relational operators.
  - Use of relational algebra operators on existing tables (relations) produces new relations.

Relational Algebra Operators (continued)

1) UNION
2) INTERSECT
3) DIFFERENCE
4) PRODUCT
5) SELECT
6) PROJECT
7) JOIN
8) DIVIDE

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

Relational Algebra Operators (continued)

Figure 1

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

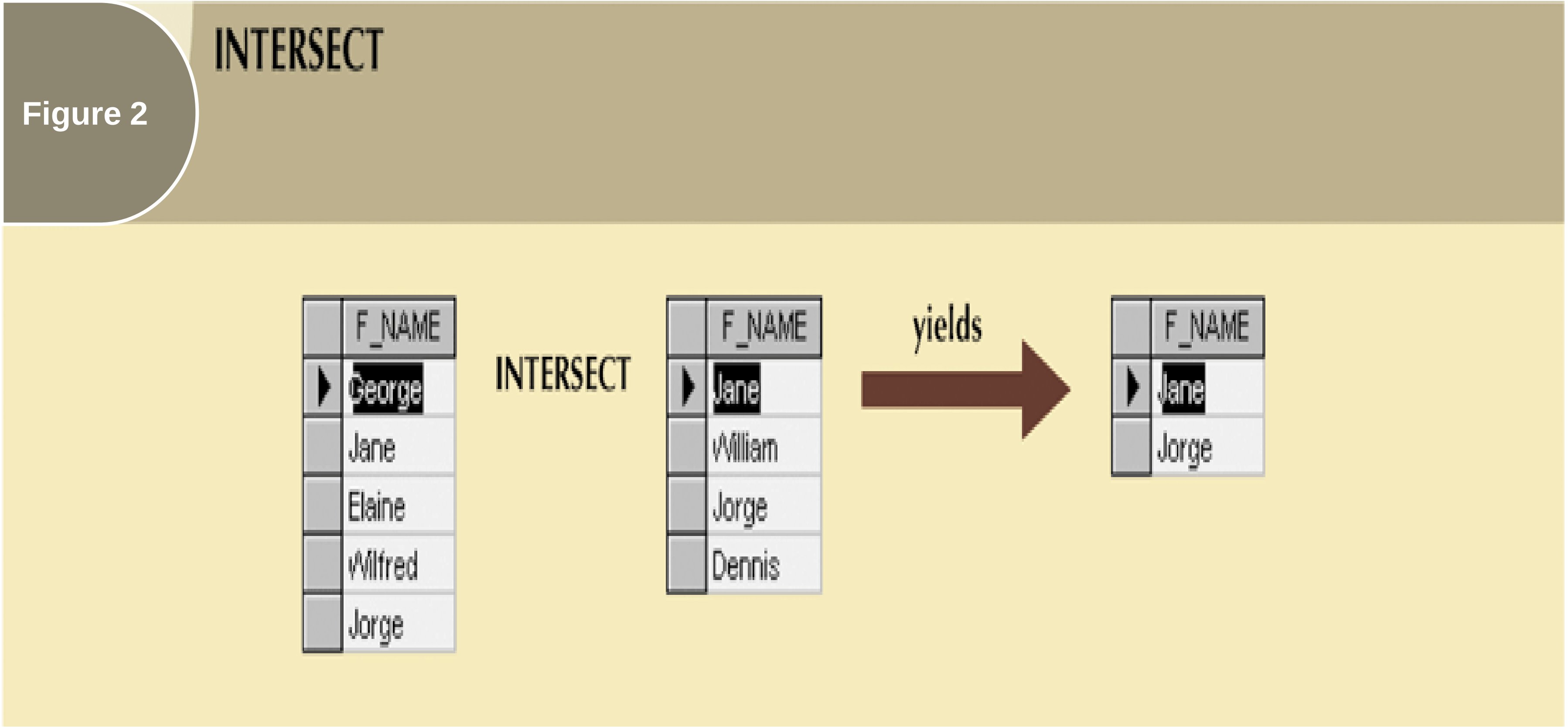
UNION

P_CODE	P_DESCRIPT	PRICE
345678	Microwave	$160.00
345679	Dishwasher	$500.00

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	$160.00
345679	Dishwasher	$500.00

Relational Algebra Operators (continued)

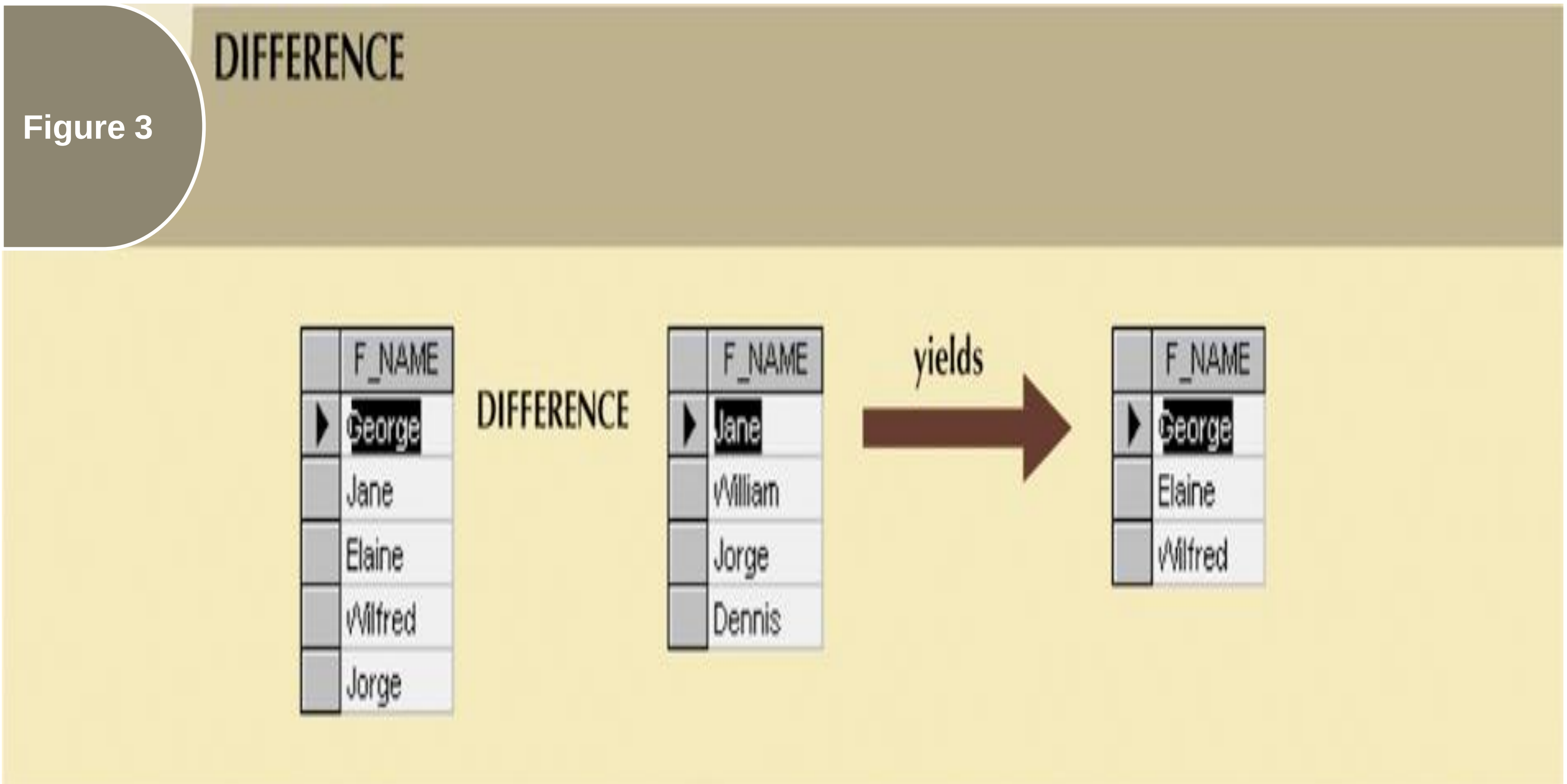


Relational Algebra Operators (continued)

- **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.

Relational Algebra Operators (continued)



Relational Algebra Operators (continued)

Figure 4

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

Relational Algebra Operators (continued)

- **Select (σ):**
  - 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 (π):**
  - Yields all values for selected attributes.
  - Yields a vertical subset of a table.

Relational Algebra Operators (continued)

Figure 5 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 or list

	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

Relational Algebra Operators (continued)

Figure 6

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 or list

	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

Relational Algebra Operators (continued)

- **Join (⋈):**
  - Allows information to be combined from two or more tables.
  - Real power behind the relational database, allowing the use of independent tables linked by common attributes.

Relational Algebra Operators (continued)

FIGURE Two tables that will be used in join illustrations

Figure 7

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

Relational Algebra Operators (continued)

- **Natural Join (⋈):**
  - Links tables by selecting only rows with common values in their common attribute(s).

  - Result of a three-stage process:
    - PRODUCT of the tables is created.
    - 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)
    - PROJECT is performed on Step 2 results to yield a single copy of each attribute, thereby eliminating duplicate columns

Relational Algebra Operators (continued)

Figure 8

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

Relational Algebra Operators (continued)

Figure 9

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

Relational Algebra Operators (continued)

Figure 10

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

Relational Algebra Operators (continued)

- **Natural Join:**
  - 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.

Relational Algebra Operators (continued)

- **Natural Join (continued):**
  - The column on which the join was made - 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.

Relational Algebra Operators (continued)

- **Equijoin (⋈_A):**
  - 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 (⋈θ):**
  - If any other comparison operator is used.

Relational Algebra Operators (continued)

- Outer join (⟕ outer left or ⟖ right outer join or ⟗ full 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

Relational Algebra Operators (continued)

Figure 11

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	

Relational Algebra Operators (continued)

Figure 12

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

Relational Algebra Operators (continued)

- **DIVIDE (÷):**
  - Requires the use of one single-column table and one two-column table.
  - Suppose we have two relations R and S with the following schemas and tuples:

    R(A, B) = { (1, 2), (2, 4), (3, 6), (4, 8) }

    S(B) = { 2, 4 }

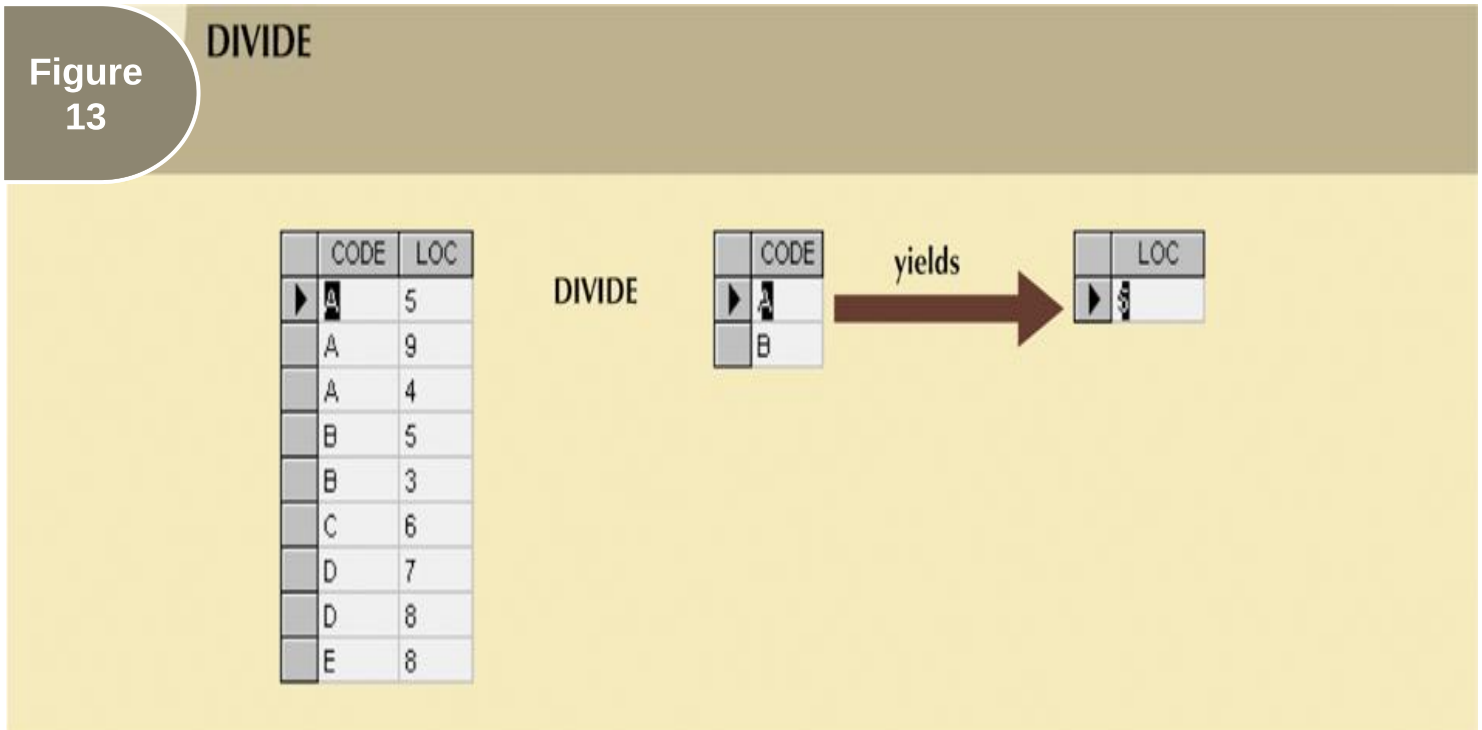
Relational Algebra Operators (continued)

- **DIVIDE (÷):**
  - To compute the divide operation R ÷ S, we need to find all tuples in R that are related to all tuples in S. In this case, we need to find tuples in R that have the values 2 and 4 in the B attribute.

  - The resulting relation has only one attribute (A), which is the attribute of R that is not in S.

    R(A,) = { (1,), (2,)}

Relational Algebra Operators (continued)



CODE	LOC
A	5
A	9
A	4
B	5
B	3
C	6
D	7
D	8
E	8

DIVIDE

CODE
A
B

yields

LOC
5

The Data Dictionary and System Catalog

- Data dictionary
  - Provides 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

Figure 14

A Sample Data Dictionary

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

Note: Telephone area codes are always composed of digits 0–9. Because area codes are not used arithmetically, they are most efficiently stored as character data. Also, the area codes are always composed of three digits. Therefore, the area code data type is defined as CHAR(3). On the other hand, names do not conform to some standard length. Therefore, the customer first names are defined as VARCHAR(20), thus indicating that up to 20 characters may be used to store the names. Character data are shown as left-justified.

The Data Dictionary and 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.

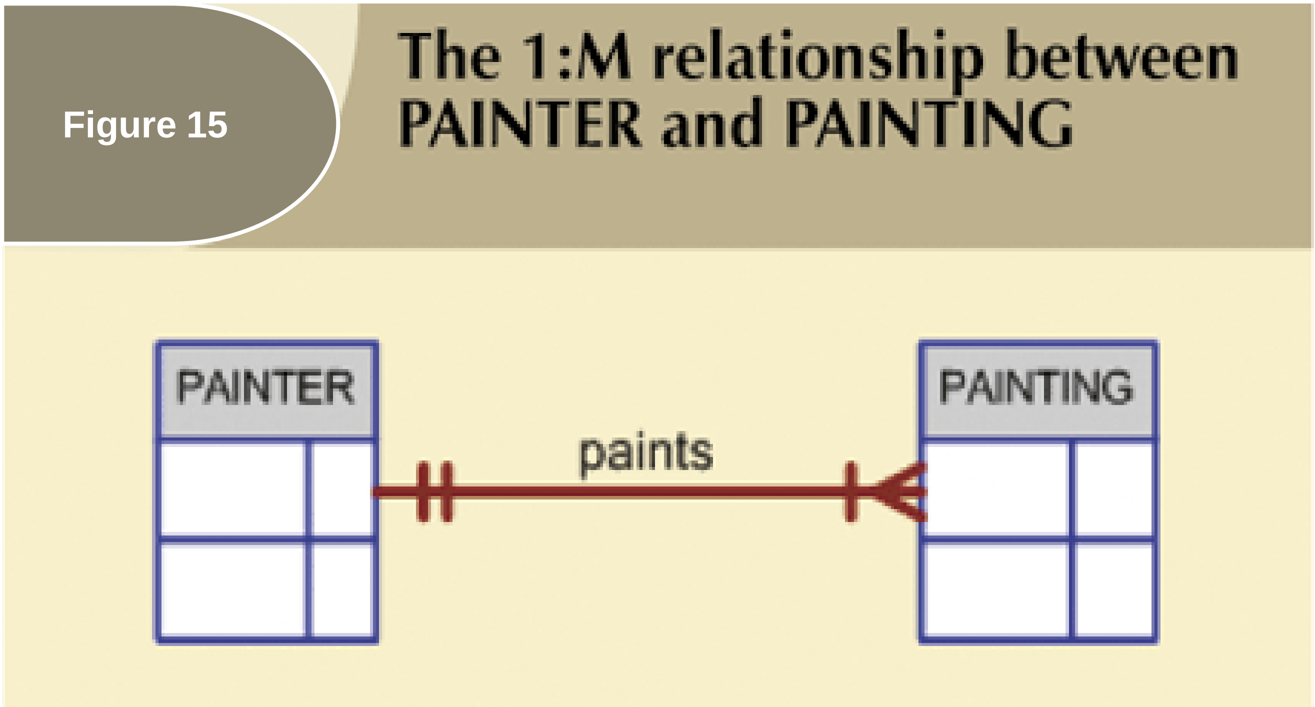
Relationships within the Relational Database

- 1:M relationship:
  - Relational modeling ideal.
  - Should be the norm in any relational database design.
- 1:1 relationship:
  - Should be rare in any relational database design
- M:N relationships:
  - Cannot be implemented as such in the relational model
  - M:N relationships can be changed into two 1:M relationships

The 1:M Relationship

- Relational database norm.
- Found in any database environment.

The 1:M Relationship (continued)



The 1:M Relationship (continued)

Figure 16 The implemented 1:M relationship between PAINTER and PAINTING

Table name: PAINTER
Primary key: PAINTER_NUM
Foreign key: none

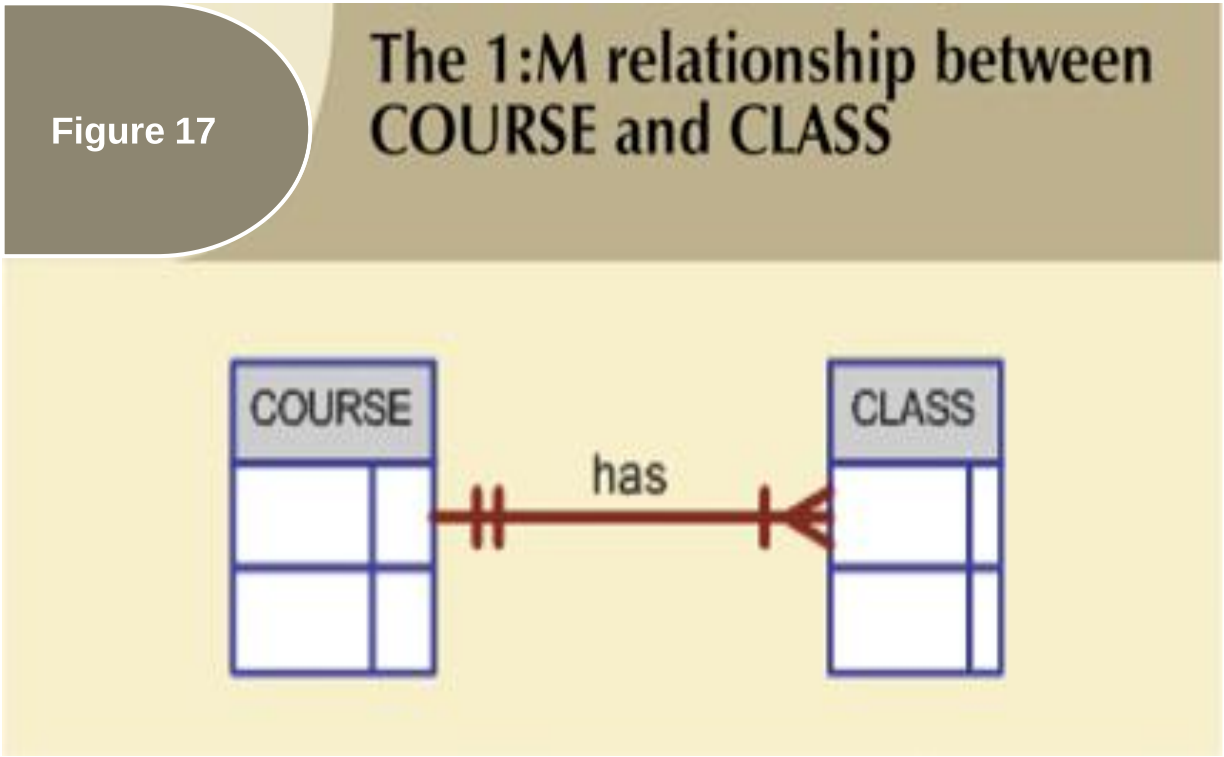
Database name: Ch03_Museum

PAINTER_NUM	PAINTER_LNAME	PAINTER_FNAME	PAINTER_INITIAL
123	Ross	Georgette	P
126	Itero	Julio	G

Table name: PAINTING
Primary key: PAINTING_NUM
Foreign key: PAINTER_NUM

PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126

The 1:M Relationship (continued)



The 1:M Relationship (continued)

Figure 18

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

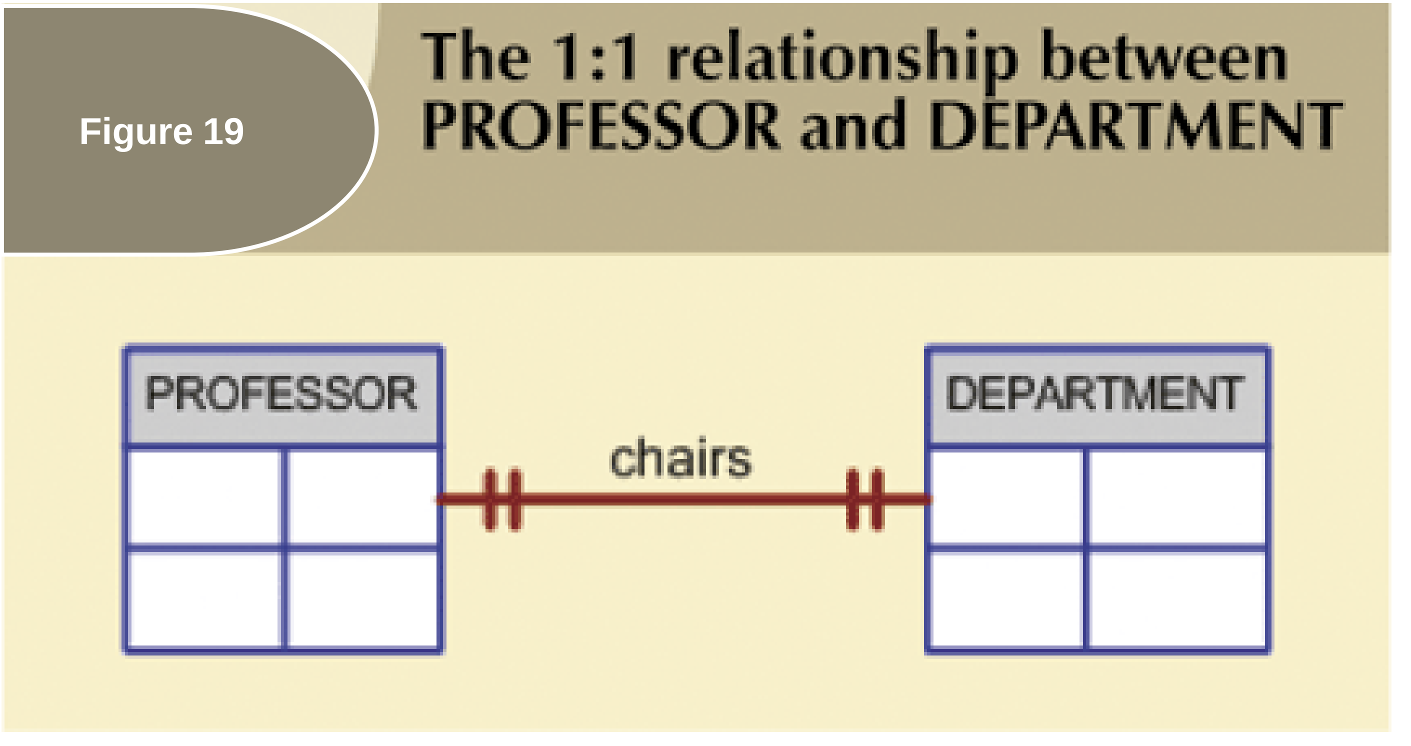
Table name: CLASS
Primary key: CLASS_CODE
Foreign key: CRS_CODE

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 1:1 Relationship

- One entity can be related to only one other entity, and vice versa.
- Sometimes means that entity components were not defined properly.
- Could indicate that two entities actually belong in the same table.
- As rare as 1:1 relationships should be, certain conditions absolutely require their use.

The 1:1 Relationship (continued)



The 1:1 Relationship (continued)

Figure 20 The implemented 1:1 relationship between PROFESSOR and DEPARTMENT

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

Database name: Ch03_TinyCollege

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.

The 1:M DEPARTMENT employs PROFESSOR relationship is implemented through the placement of the DEPT_CODE foreign key in the PROFESSOR table.

The 1:1 PROFESSOR chairs DEPARTMENT relationship is implemented through the placement of the EMP_NUM foreign key in the DEPARTMENT table.

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

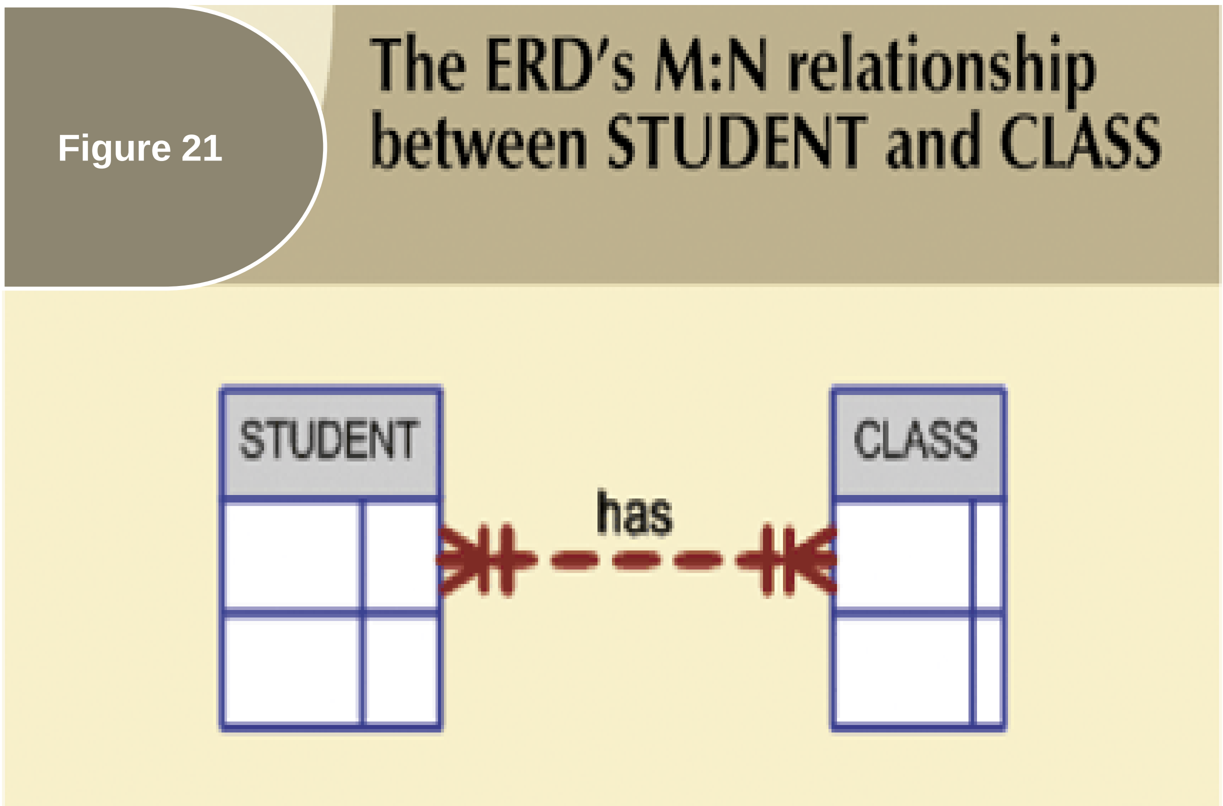
DEPT_CODE	DEPT_NAME	SCHOOL_CODE	EMP_NUM	DEPT_ADDRESS	DEPT_EXTENSION
ACCT	Accounting	BUS	114	KLR 211, Box 52	3119
ART	Fine Arts	A&SCI	435	BBG 185, Box 128	2278
BIOL	Biology	A&SCI	387	AAK 230, Box 415	4117
CIS	Computer Info. Systems	BUS	209	KLR 333, Box 56	3245
ECON/FIN	Economics/Finance	BUS	299	KLR 284, 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 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 M:N Relationship (continued)



The M:N Relationship (continued)

Figure 22 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 (continued)

Figure 23

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

The M:N Relationship (continued)

- 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.

The M:N Relationship (continued)

Figure 24 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

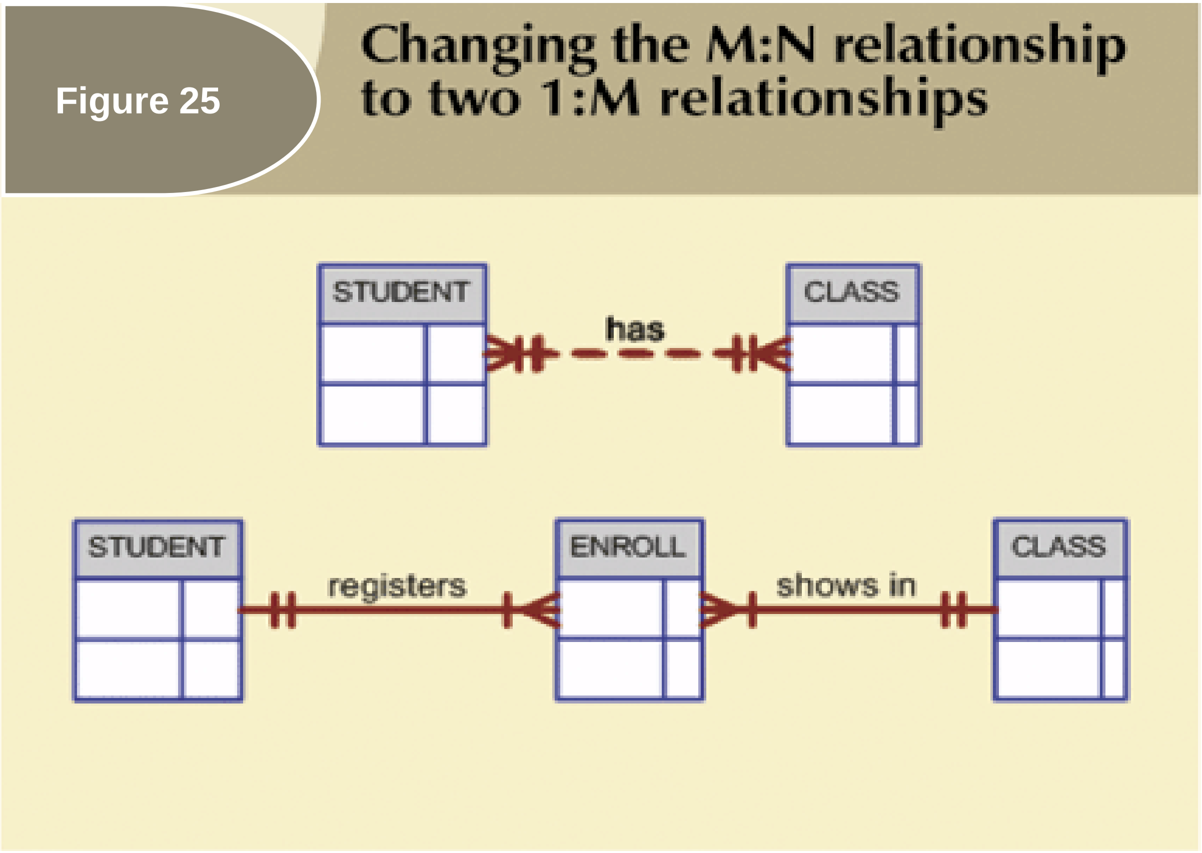
Table name: ENROLL
Primary key: CLASS_CODE + STU_NUM
Foreign key: CLASS_CODE, STU_NUM

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

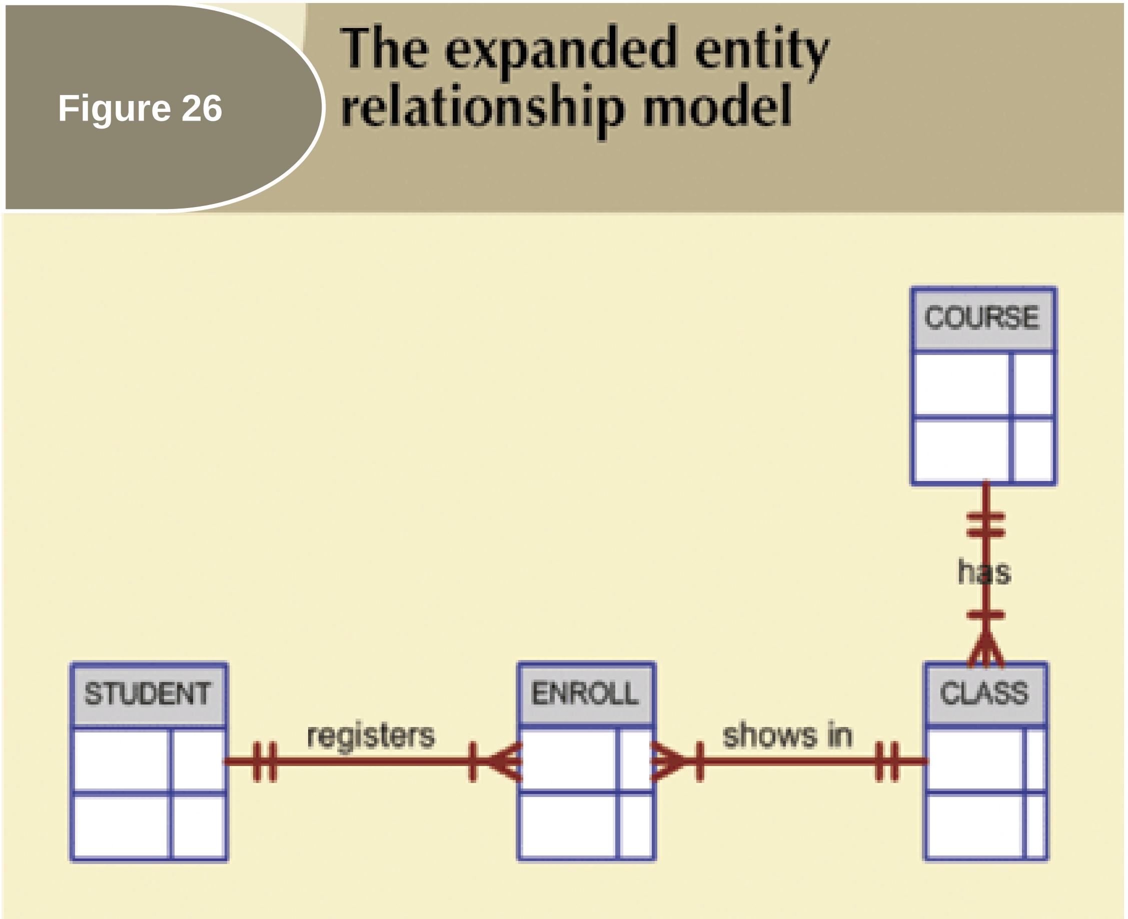
Table name: CLASS
Primary key: CLASS_CODE
Foreign key: CRS_CODE

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

The M:N Relationship (continued)



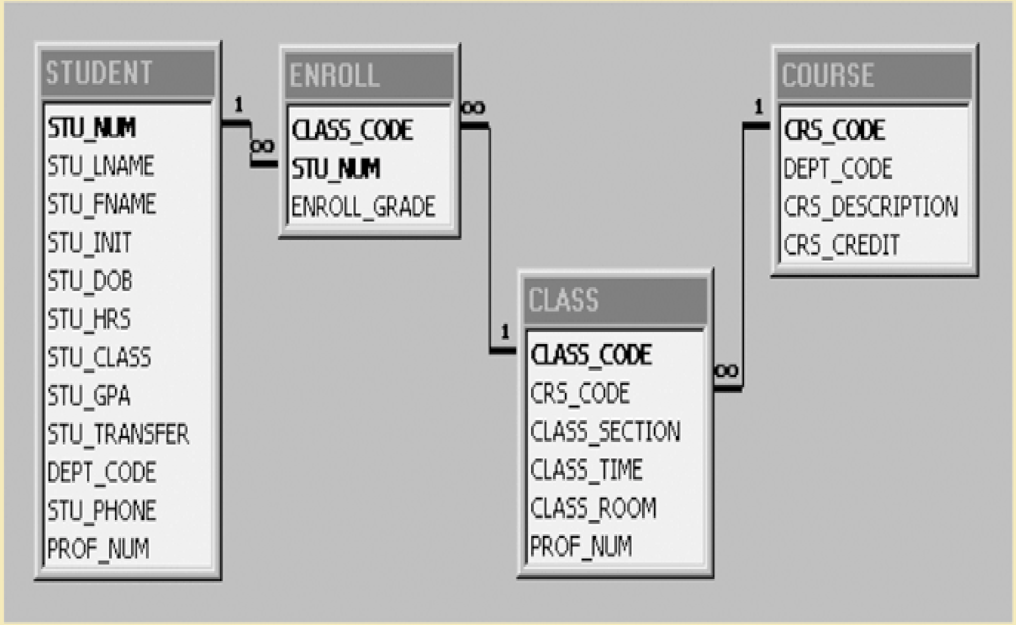
The M:N Relationship (continued)



The M:N Relationship (continued)

Figure 27

The relational diagram for the Ch03_TinyCollege database



Exercise

- **Question One:**
  - Write the example SQL statements for all the Relational algebra that we have discussed in this lecture.
- **Question Two:**
  - Find other relational algebra and with example write the SQL statement for each of the relational algebra you found.

END!!!!