Database Systems: Design, Implementation, and Management

Lesson 3

Objectives

- In this lesson, students will learn:
  - That the relational database model offers a logical view of data
  - About the relational model's basic component: relations
  - That relations are logical constructs composed of rows (tuples) and columns (attributes)
  - That relations are implemented as tables in a relational DBMS

Objectives (cont'd.)

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

A Logical View of Data

- Relational model
  - View data logically rather than physically
- Table
  - Structural and data independence
  - Resembles a file conceptually
- Relational database model is easier to understand than hierarchical and network models

Tables and Their Characteristics

- Logical view of relational database is based on relation
  - Relation thought of as a table
- Table: two-dimensional structure composed of rows and columns
  - Persistent representation of logical relation
- Contains group of related entities (entity set)

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.
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 a combination of attributes that uniquely identifies each row.

FIGURE 3.1 STUDENT table attribute values

Table name: STUDENT **Database name: Ch03_TinyCollege**

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
321452	Bowser	William	C	12-Feb-1975	42	So	2.84	No	BIOL	2134	205
324257	Smithson	Anne	K	15-Nov-1981	81	Jr	3.27	Yes	CIS	2256	222
324258	Brewer	Juliette		23-Aug-1969	36	So	2.26	Yes	ACCT	2256	228
324269	Oblonski	Walter	H	16-Sep-1976	66	Jr	3.09	No	CIS	2114	222
324273	Smith	John	D	30-Dec-1958	102	Sr	2.11	Yes	ENGL	2231	199
324274	Katinga	Raphael	P	21-Oct-1979	114	Sr	3.15	No	ACCT	2267	228
324291	Robertson	Gerald	T	08-Apr-1973	120	Sr	3.87	No	EDU	2267	311
324299	Smith	John	B	30-Nov-1986	15	Fr	2.92	No	ACCT	2315	230

STU_NUM = Student number
STU_LNAME = Student last name
STU_FNAME = Student first name
STU_INIT = Student middle initial
STU_DOB = Student date of birth
STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_GPA = Grade point average
STU_TRANSFER = Student transferred from another institution
DEPT_CODE = Department code
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

Keys

- Each row in a table must be uniquely identifiable
- **Key** is one or more attributes that determine other attributes
- Key's role is based on **determination**
  - If you know the value of attribute A, you can determine the value of attribute B

$$\text{STU\_NUM} \rightarrow \text{STU\_LNAME, STU\_FNAME, STU\_INIT}$$

The Stu_Num value in the STUDENT determines all the student's attribute values

Keys (cont'd.)

- **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 row
- **Candidate key**
  - A superkey without unnecessary attributes

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

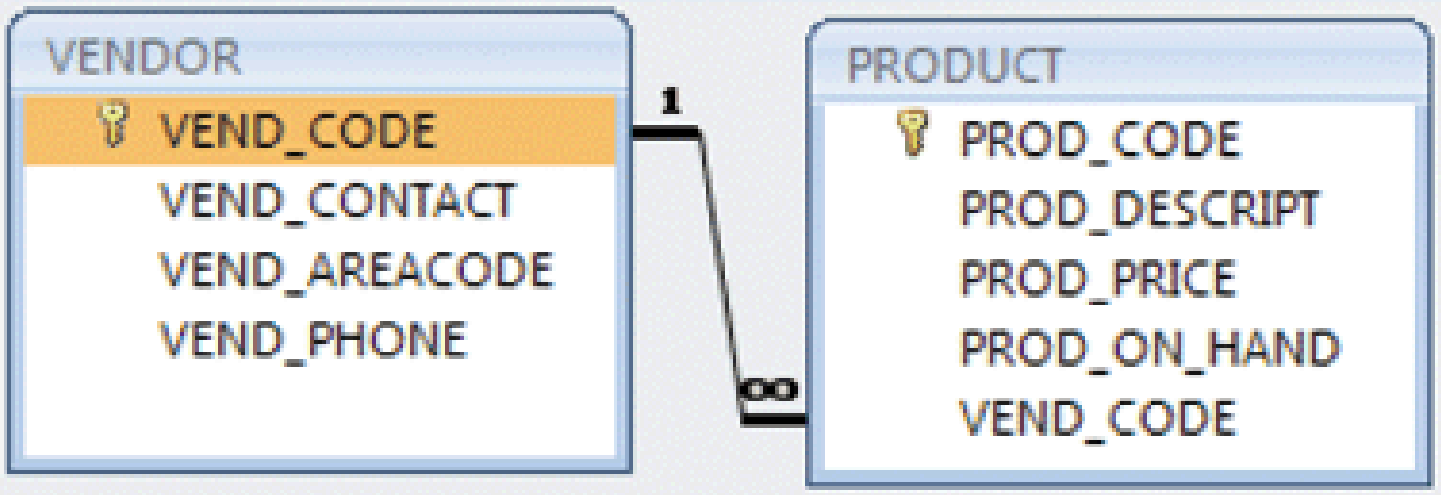
PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

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

FIGURE 3.3 The relational diagram for the Ch03_SaleCo database



Keys (cont'd.)

- **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

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.
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

- Many RDBMs enforce integrity rules automatically
- Safer to ensure that application design conforms to entity and referential integrity rules

FIGURE 3.4 An illustration of integrity rules

Table name: CUSTOMER
Primary key: CUS_CODE
Foreign key: AGENT_CODE

Database name: Ch03_InsureCo

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_INSURE_TYPE	CUS_INSURE_AMT	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	T1	100.00	05-Apr-2010	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2010	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2011	502
10013	Olowski	Paul	F	615	894-2180	S1	300.00	14-Oct-2010	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2010	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2010	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2011	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2010	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2010	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2011	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	132735.75
502	615	882-1244	Hahn	138967.35
503	615	123-5589	Okon	127093.45

The Data Dictionary

- **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

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 account code	CHAR(5)	99999	10000–99999	Y	PK	
	CUS_LNAME	Customer last name	VARCHAR(20)	Xxxxxxxx		Y		
	CUS_FNAME	Customer first name	VARCHAR(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_CODE
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		Y		
	AGENT_LNAME	Agent last name	VARCHAR(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)) are 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.

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

Relationships within the Relational Database (cont'd.)

- M:N relationships
  - Cannot be implemented as such in the relational model
  - M:N relationships can be changed into 1:M relationships

The 1:M Relationship

- Relational database norm
- Found in any database environment

FIGURE 3.18 The 1:M relationship between PAINTER and PAINTING

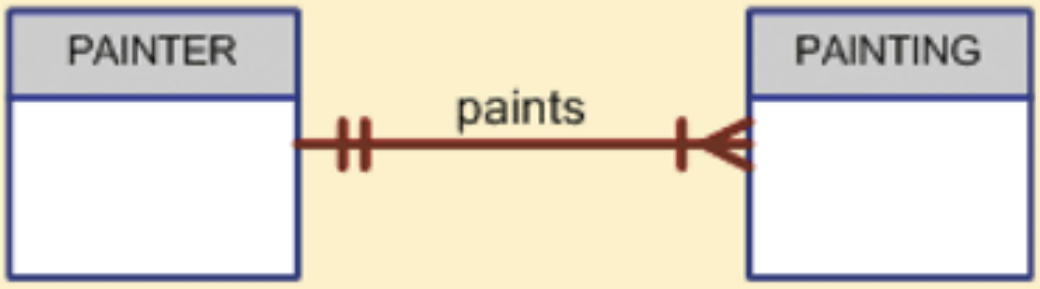


FIGURE 3.19 The implemented 1:M relationship between PAINTER and PAINTING

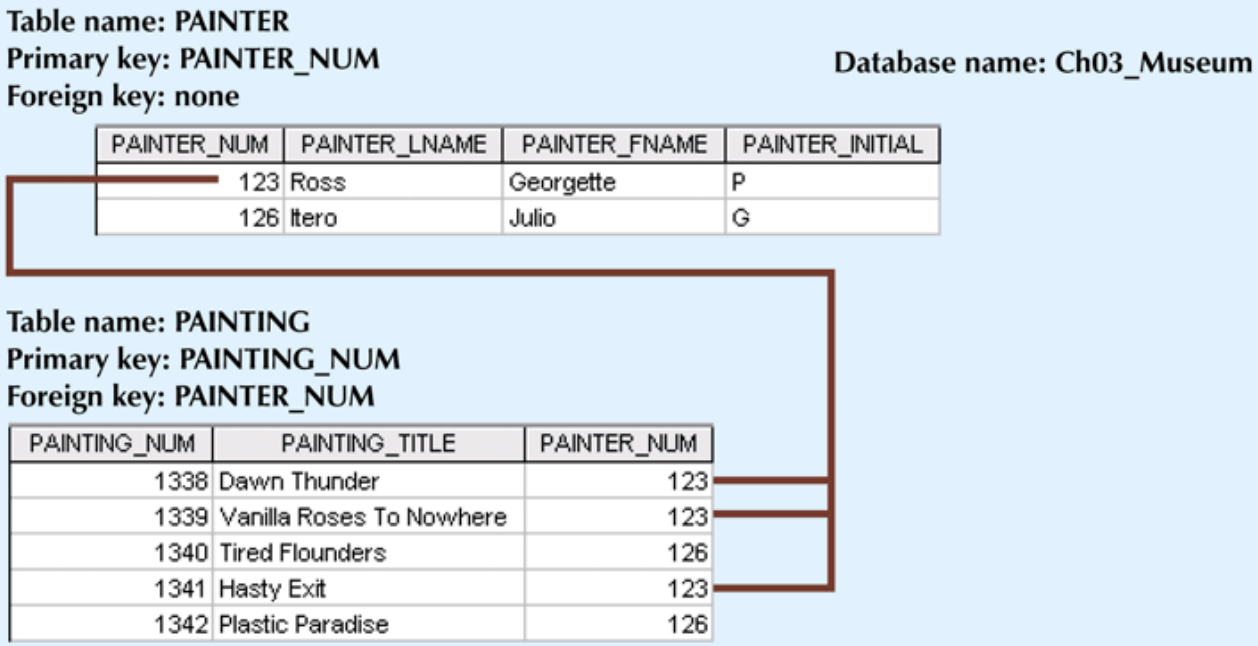


Table name: PAINTER
Primary key: PAINTER_NUM
Foreign key: none

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

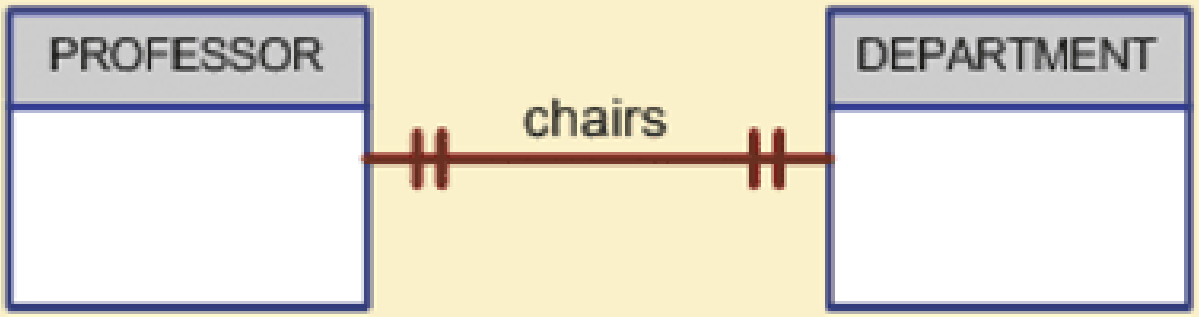
PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126

The 1:1 Relationship

- One entity 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
- Certain conditions absolutely require their use

FIGURE 3.22

The 1:1 relationship between PROFESSOR and DEPARTMENT



The M:N Relationship

- Implemented by breaking it up to produce a set of 1:M relationships
- Avoid problems inherent to M:N relationship by creating a **composite entity**
  - Includes as foreign keys the primary keys of tables to be linked

FIGURE 3.24 The ERM's M:N relationship between STUDENT and CLASS

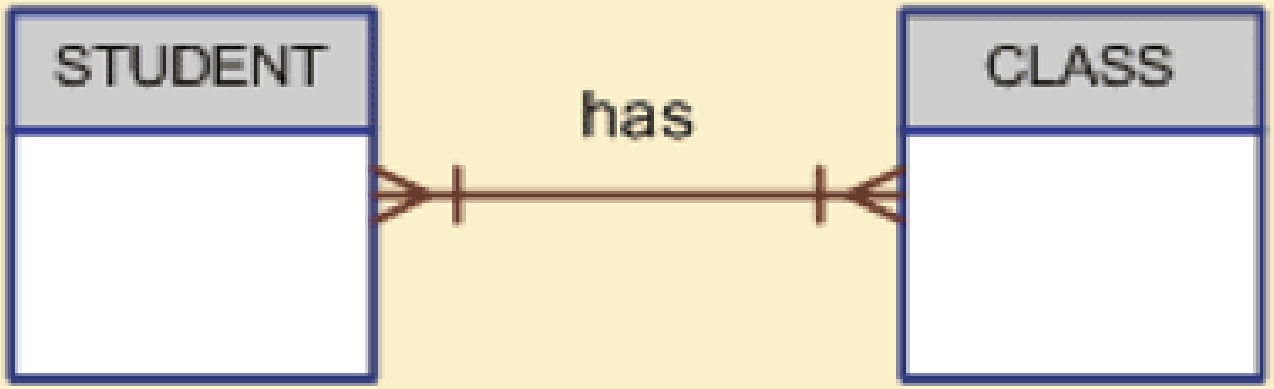


FIGURE 3.26 Converting the M:N relationship into two 1:M relationships

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

FIGURE 3.27 Changing the M:N relationship to two 1:M relationships

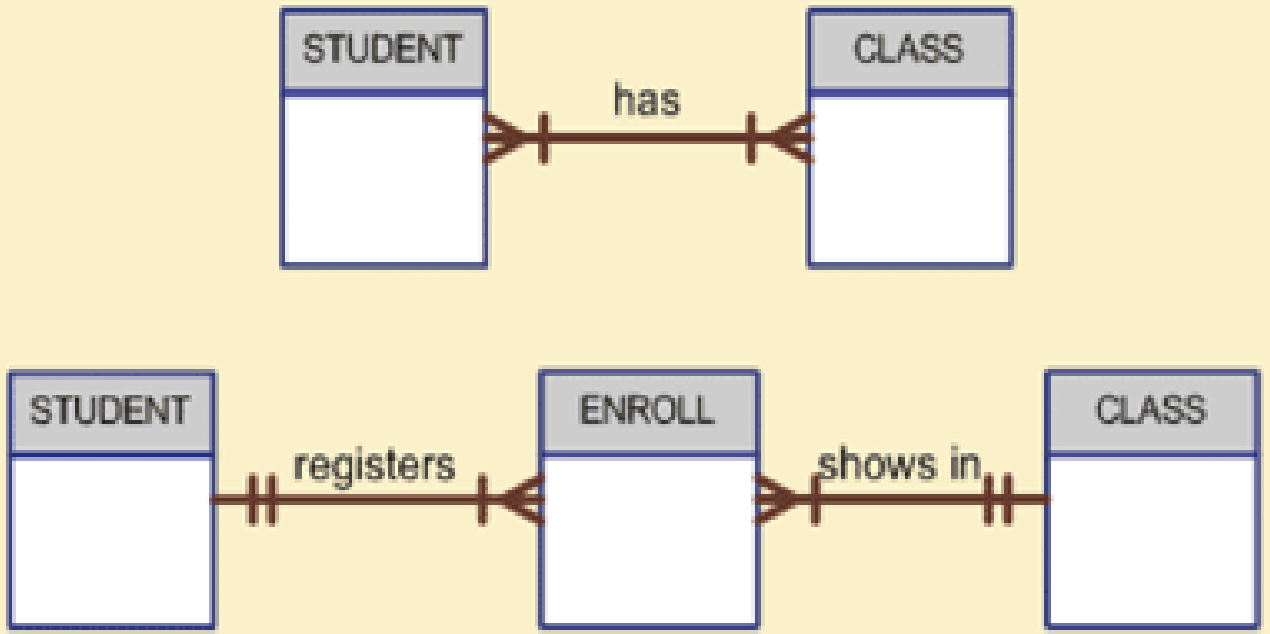


FIGURE 3.28 The expanded entity relationship model

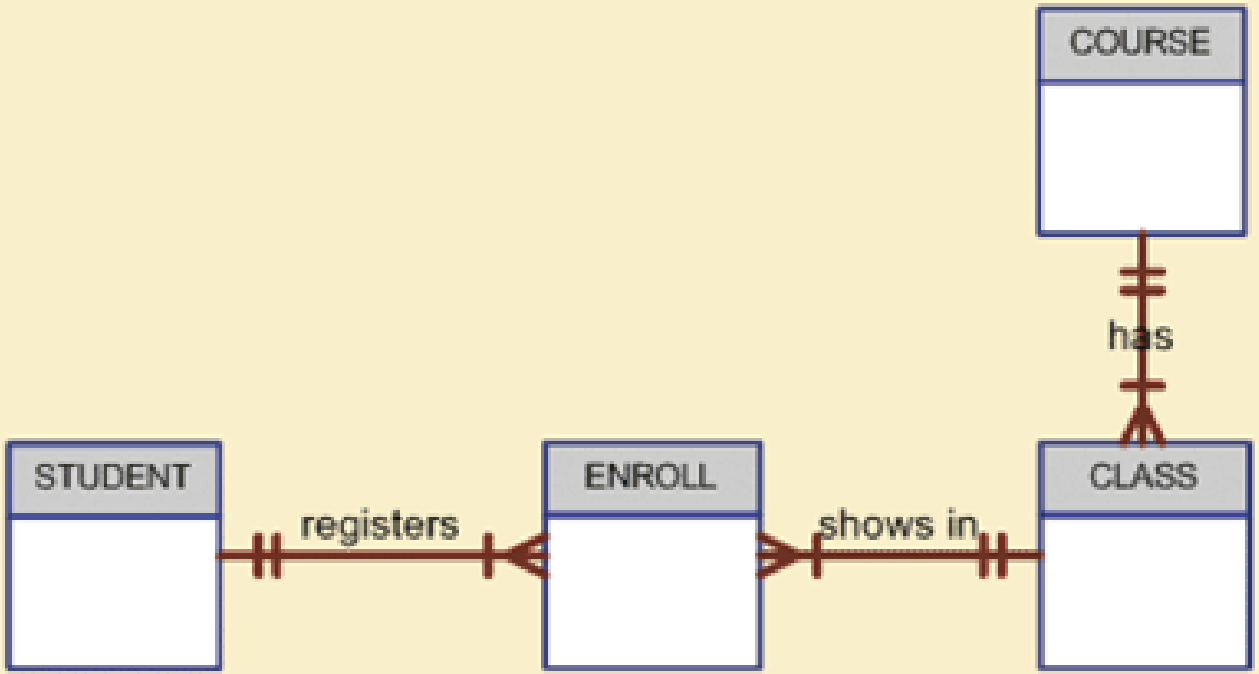


FIGURE 3.29 The relational diagram for the Ch03_TinyCollege database

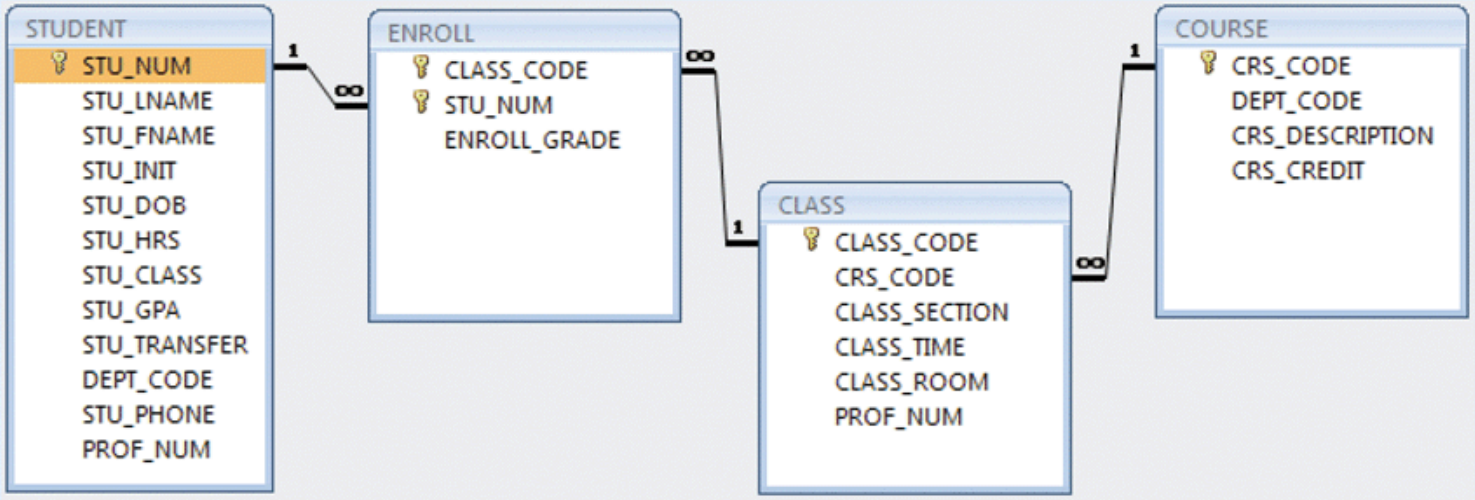
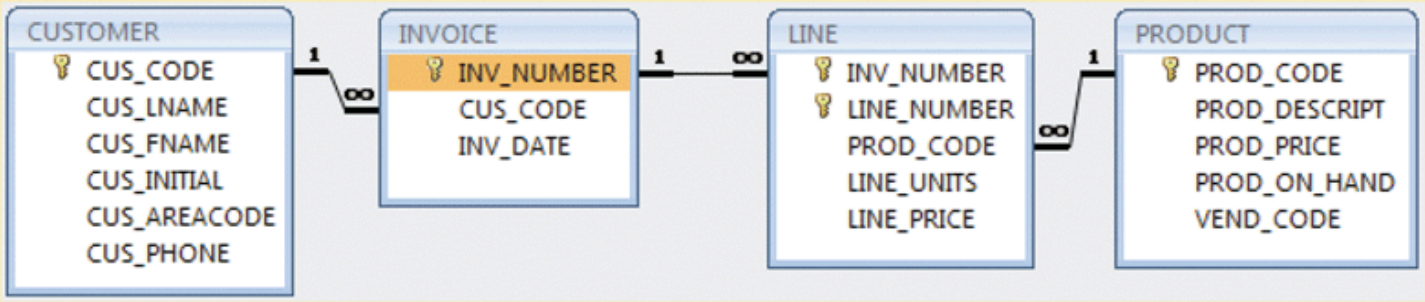


FIGURE 3.31 The relational diagram for the invoicing system



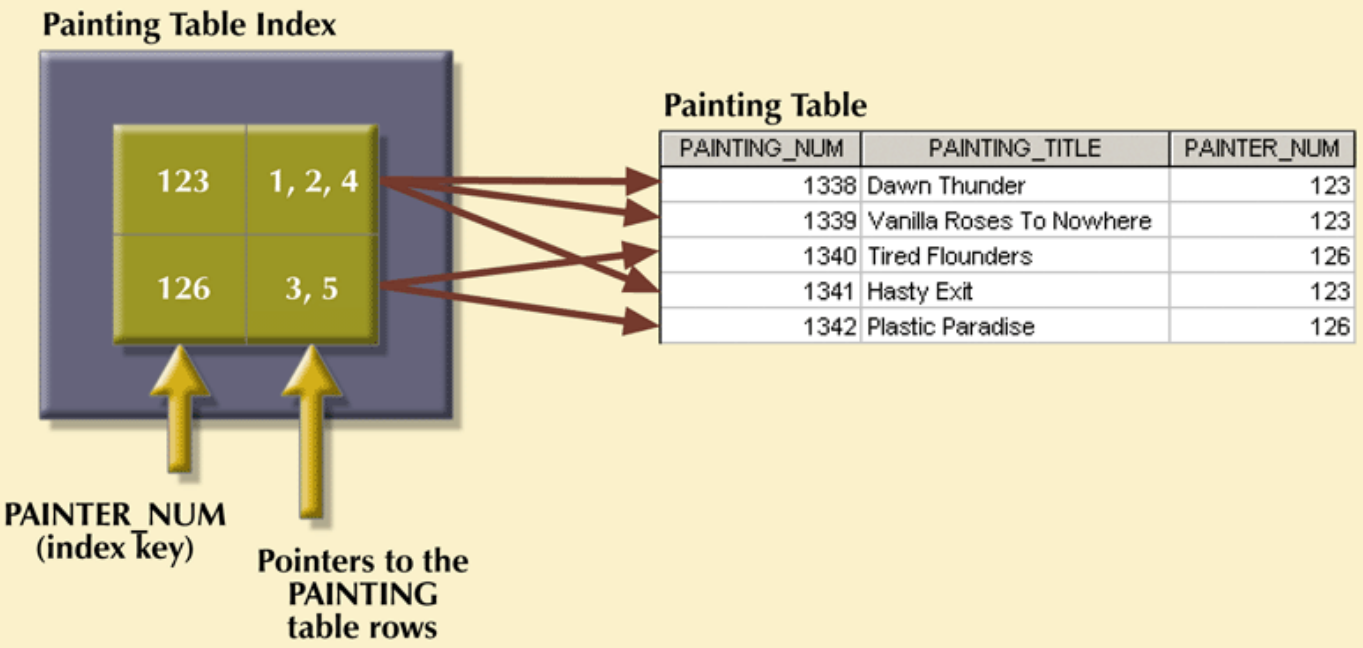
Data Redundancy Revisited

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

Indexes

- Orderly arrangement 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 have only one pointer value (row) associated with it
- Each index is associated with only one table

FIGURE 3.32 Components of an index



Painting Table

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