The National Higher School of Artificial Intelligence

DATABASES

Chapter 2 : The Relational Database Model

Pr. Kamel BOUKHALFA

16/10/2022

Slides From the Textbook :
Carlos Coronel and Steven Morris, Database Systems: Design, Implementation, and Management
Tenth Edition

Objectives

In this chapter, 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
- Why indexing is important

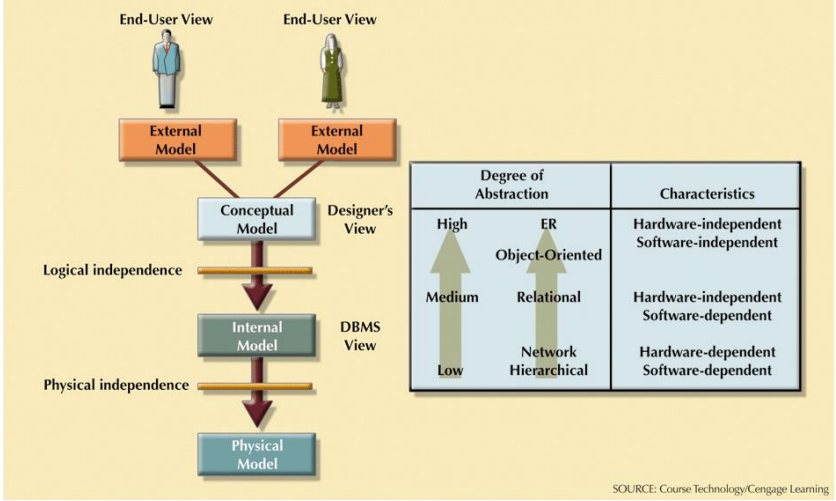
Degrees of Data Abstraction

- Database designer starts with abstracted view, then adds details
- ANSI Standards Planning and Requirements Committee (SPARC)
  - Defined a framework for data modeling based on degrees of data abstraction (1970s):
    - External
    - Conceptual
    - Internal

The External Model

- End users' view of the data environment
- ER diagrams represent external views
- External schema: specific representation of an external view
  - Entities
  - Relationships
  - Processes
  - Constraints

FIGURE 2.7 Data abstraction levels



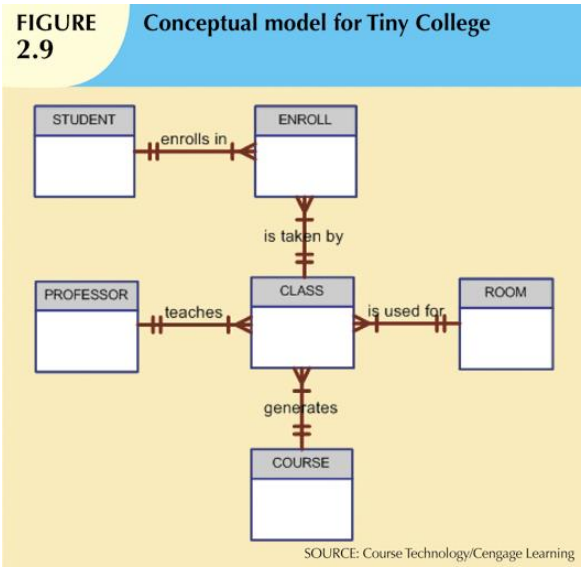
The External Model (cont'd.)

- Easy to identify specific data required to support each business unit's operations
- Facilitates designer's job by providing feedback about the model's adequacy
- Ensures security constraints in database design
- Simplifies application program development

The Conceptual Model

- Represents global view of the entire database
- All external views integrated into single global view: conceptual schema
- ER model most widely used
- ERD graphically represents the conceptual schema

FIGURE 2.9 Conceptual model for Tiny College



SOURCE: Course Technology/Cengage Learning

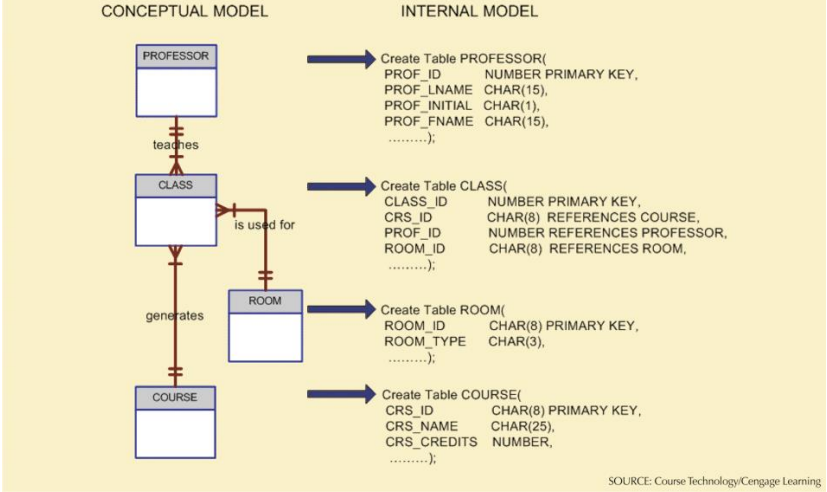
The Conceptual Model (cont'd.)

- Provides a relatively easily understood macro level view of data environment
- Independent of both software and hardware
  - Does not depend on the DBMS software used to implement the model
  - Does not depend on the hardware used in the implementation of the model
  - Changes in hardware or software do not affect database design at the conceptual level

The Internal Model

- Representation of the database as “seen” by the DBMS
  - Maps the conceptual model to the DBMS
- Internal schema depicts a specific representation of an internal model
- Depends on specific database software
  - Change in DBMS software requires internal model be changed
- Logical independence: change internal model without affecting conceptual model

FIGURE 2.10 Internal model for Tiny College



SOURCE: Course Technology/Cengage Learning

The Physical Model

- Operates at lowest level of abstraction
  - Describes the way data are saved on storage media such as disks or tapes
- Requires the definition of physical storage and data access methods
- Relational model aimed at logical level
  - Does not require physical-level details
- Physical independence: changes in physical model do not affect internal model

TABLE 2.4 Levels of Data Abstraction

MODEL	DEGREE OF ABSTRACTION	FOCUS	INDEPENDENT OF
External	High	End-user views	Hardware and software
Conceptual	↕	Global view of data (database model independent)	Hardware and software
Internal	↕	Specific database model	Hardware
Physical	Low	Storage and access methods	Neither hardware nor software

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 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.

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

SOURCE: Course Technology/Cengage Learning

Keys

- Each row in a table must be uniquely identifiable
- Key: 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
  - Functional dependence
    - Attribute B is functionally dependent on A if all rows in table that agree in value for A also agree in value for B

Types of Keys

- 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

Types of Keys (cont'd.)

- Entity integrity
  - Each row (entity instance) in the table has its own unique identity
- Nulls
  - No data entry
  - Not permitted in primary key
  - Should be avoided in other attributes

Types of Keys (cont'd.)

- Can represent:
  - An unknown attribute value
  - A known, but missing, attribute value
  - A "not applicable" condition
- Can create problems when functions such as COUNT, AVERAGE, and SUM are used
- Can create logical problems when relational tables are linked

Types of Keys (cont'd.)

- Controlled redundancy
  - Makes the relational database work
  - Tables within the database share common attributes
    - Enables tables to be linked together
  - Multiple occurrences of values not redundant when required to make the relationship work
  - Redundancy exists only when there is unnecessary duplication of attribute values

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

SOURCE: Course Technology/Cengage Learning

Types of 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
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

Integrity Rules

- Many RDBMs enforce integrity rules automatically
- Safer to ensure that application design conforms to entity and referential integrity rules
- Designers use flags to avoid nulls
  - Flags indicate absence of some value

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	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 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	It is 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).

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-2012	502
10011	Dunne	Leona	K	16-Jun-2012	501
10012	Smith	Kathy	W	29-Jan-2013	502
10013	Olowski	Paul	F	14-Oct-2012	
10014	Orlando	Myron		28-Dec-2012	501
10015	O'Brian	Amy	B	22-Sep-2012	503
10016	Brown	James	G	25-Mar-2013	502
10017	Williams	George		17-Jul-2012	503
10018	Farriss	Anne	G	03-Dec-2012	501
10019	Smith	Olette	K	14-Mar-2013	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

SOURCE: Course Technology/Cengage Learning

Relational Set Operators

- Relational algebra
  - Defines theoretical way of manipulating table contents using relational operators
  - Use of relational algebra operators on existing relations produces new relations:

• SELECT	• UNION
• PROJECT	• DIFFERENCE
• JOIN	• PRODUCT
• INTERSECT	• DIVIDE

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

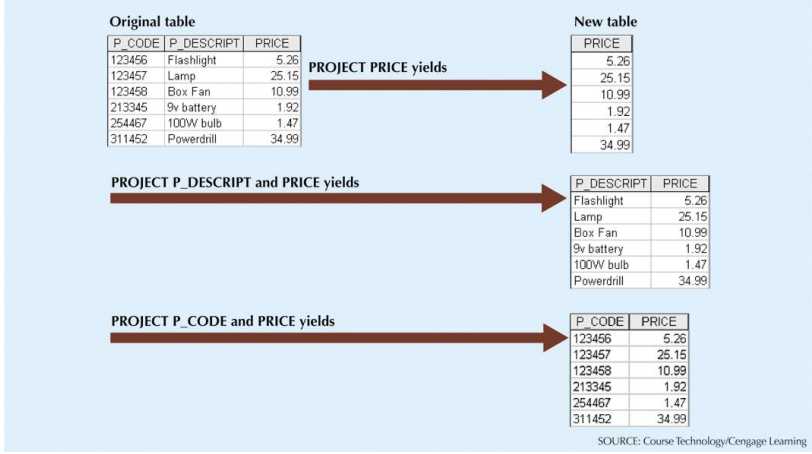
P_CODE	P_DESCRIPT	PRICE
213345	9v battery	1.92
254467	100W bulb	1.47

SELECT only P_CODE = 311452 yields

P_CODE	P_DESCRIPT	PRICE
311452	Powerdrill	34.99

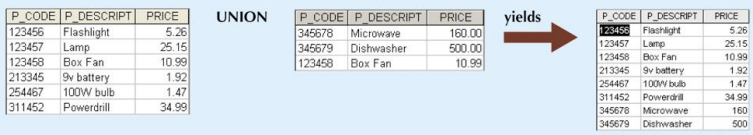
SOURCE: Course Technology/Cengage Learning

FIGURE 3.5 PROJECT



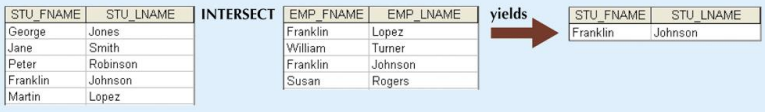
SOURCE: Course Technology/Cengage Learning

FIGURE 3.6 UNION



SOURCE: Course Technology/Cengage Learning

FIGURE 3.7 INTERSECT



SOURCE: Course Technology/Cengage Learning

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

SOURCE: Course Technology/Cengage Learning

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

SOURCE: Course Technology/Cengage Learning

Relational Set Operators (cont'd.)

- Natural join
  - Links tables by selecting rows with common values in common attributes (join columns)
- Equijoin
  - Links tables on the basis of an equality condition that compares specified columns
- Theta join
  - Any other comparison operator is used

Relational Set Operators (cont'd.)

- Inner join
  - Only returns matched records from the tables that are being joined
- Outer join
  - Matched pairs are retained, and any unmatched values in other table are left null

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

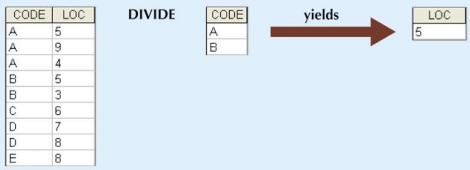
AGENT_CODE	AGENT_PHONE
125	6152439887
167	6153426778
231	6152431124
333	9041234445

SOURCE: Course Technology/Cengage Learning

Relational Set Operators (cont'd.)

- Left outer join
  - Yields all of the rows in the CUSTOMER table
  - Including those that do not have a matching value in the AGENT table
- Right outer join
  - Yields all of the rows in the AGENT table
  - Including those that do not have matching values in the CUSTOMER table

FIGURE 3.16 DIVIDE



SOURCE: Course Technology/Cengage Learning

Exercise

Actor(idA, name, Firstname, Nationality)

Film(idF, Title, Year, Country, NBSpec, *idMaker**,*idKind**)

Acting(*idActor**, *idFilm**, Salary)

Maker(idM, Name, Firstname, Nationality)

Kind(idK, Description)

41

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
- System catalog
  - Contains metadata
  - Detailed system data dictionary that describes all objects within the database

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				

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.

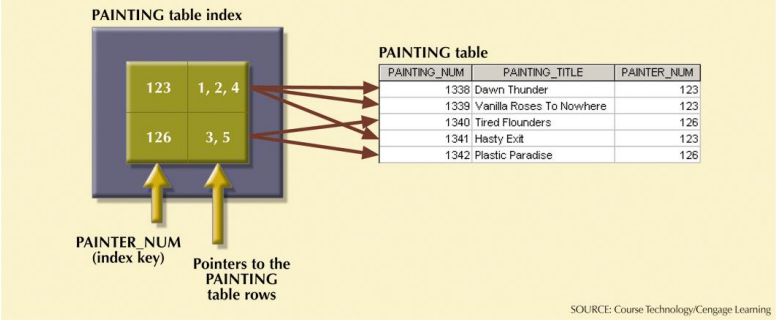
Data Redundancy Revisited

- 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 3.31 Components of an index



SOURCE: Course Technology/Cengage Learning

Summary

- Tables are basic building blocks of a relational database
- Keys are central to the use of relational tables
- Keys define functional dependencies
  - Superkey
  - Candidate key
  - Primary key
  - Secondary key
  - Foreign key

Summary (cont'd.)

- Each table row must have a primary key that uniquely identifies all attributes
- Tables are linked by common attributes
- The relational model supports relational algebra functions
  - SELECT, PROJECT, JOIN, INTERSECT UNION, DIFFERENCE, PRODUCT, DIVIDE