Database Systems: Design, Implementation, and Management

12TH Edition

Chapter 3

The Relational Database Model

Department of
Software Engineering
BAHRIA UNIVERSITY
Discovering Knowledge

The Relational Model

- Developed by E.F. Codd (IBM) in 1970
- Table (relations)
  - Matrix consisting of row/column intersections
  - Each row in a relation is called a tuple
- Relational models were considered impractical in 1970
- Model was conceptually simple at expense of computer overhead

The Relational Model (cont’d.)

- Relational data management system (RDBMS)
  - Performs same functions provided by hierarchical model
  - Hides complexity from the user
- Relational diagram
  - Representation of entities, attributes, and relationships
- Relational table stores collection of related entities

FIGURE 2.1 Linking relational tables

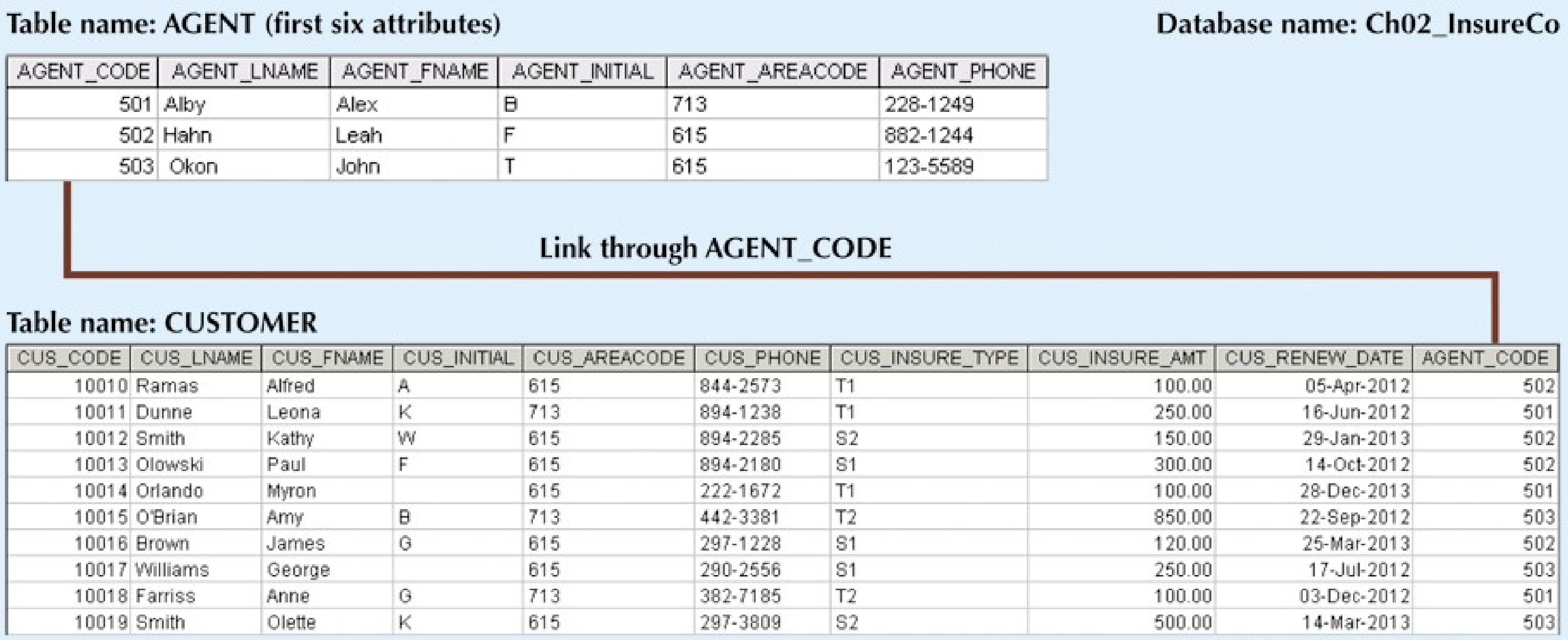


Table name: AGENT (first six attributes)

Database name: Ch02_InsureCo

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

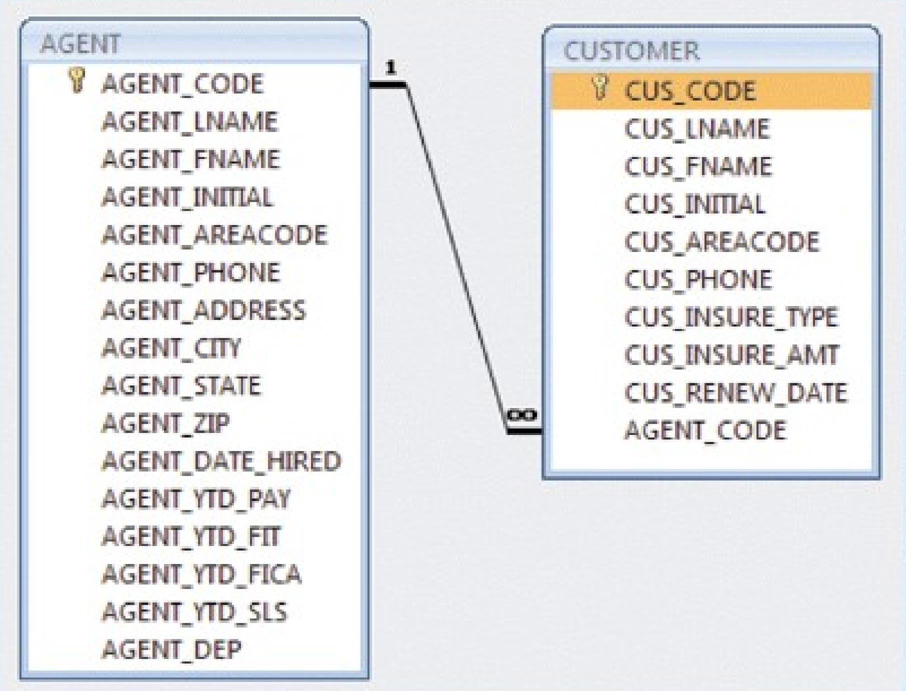
Link through AGENT_CODE

Table name: CUSTOMER

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-2012	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2012	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2013	502
10013	Olowski	Paul	F	615	894-2180	S1	300.00	14-Oct-2012	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2013	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2012	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2013	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2012	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2012	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2013	503

SOURCE: Course Technology/Cengage Learning

FIGURE 2.2 A relational diagram



SOURCE: Course Technology/Cengage Learning

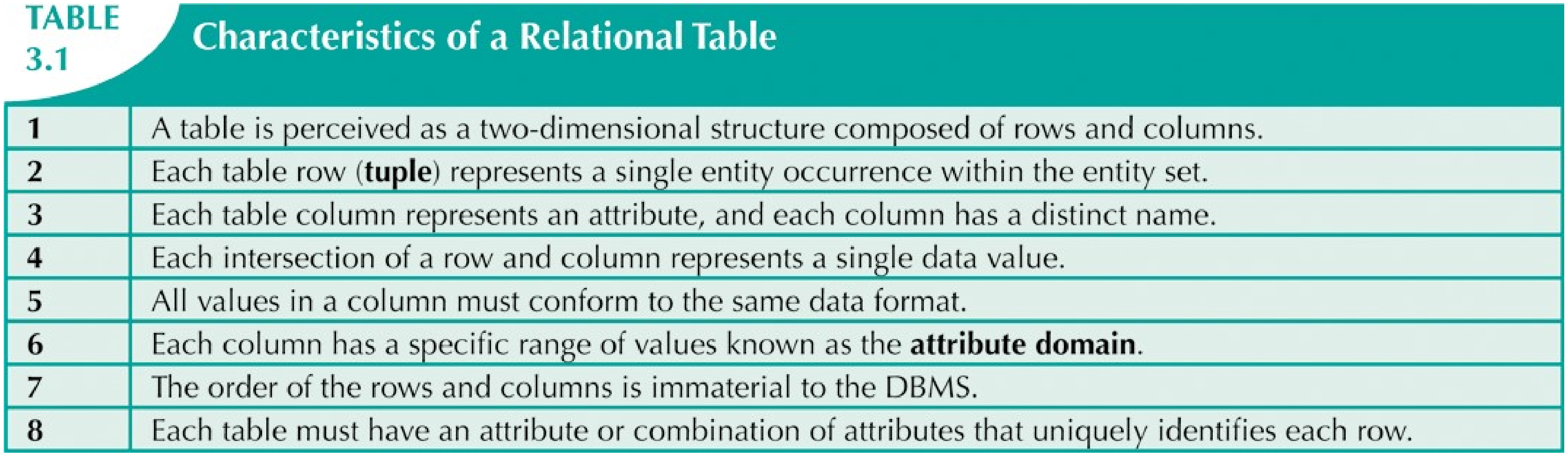
A Logical View of Data

- 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.

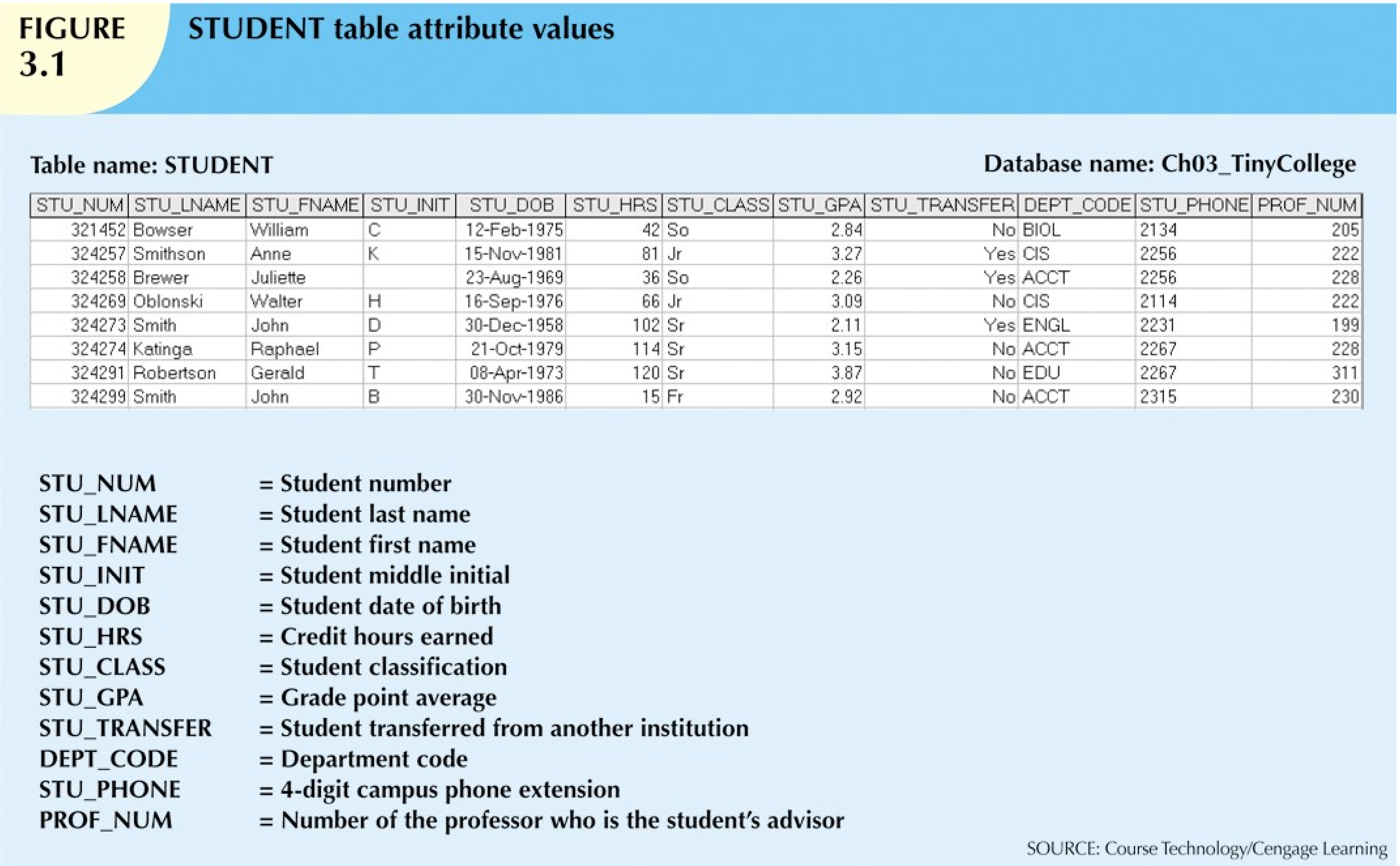


FIGURE 3.1

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

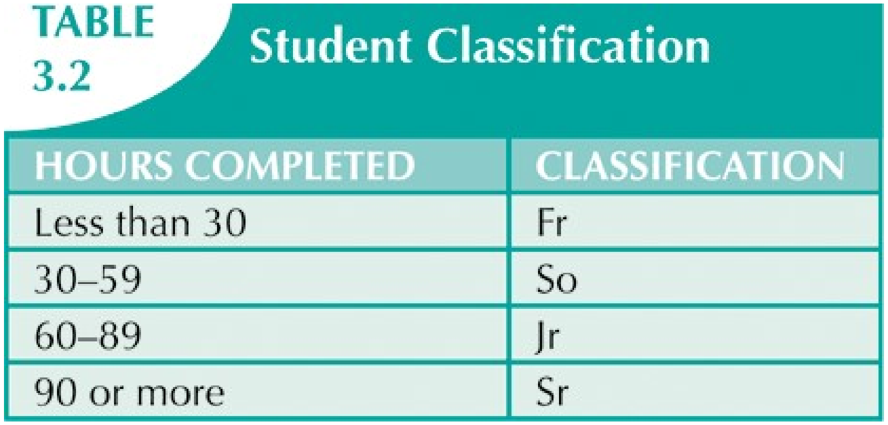


TABLE 3.2 Student Classification

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

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

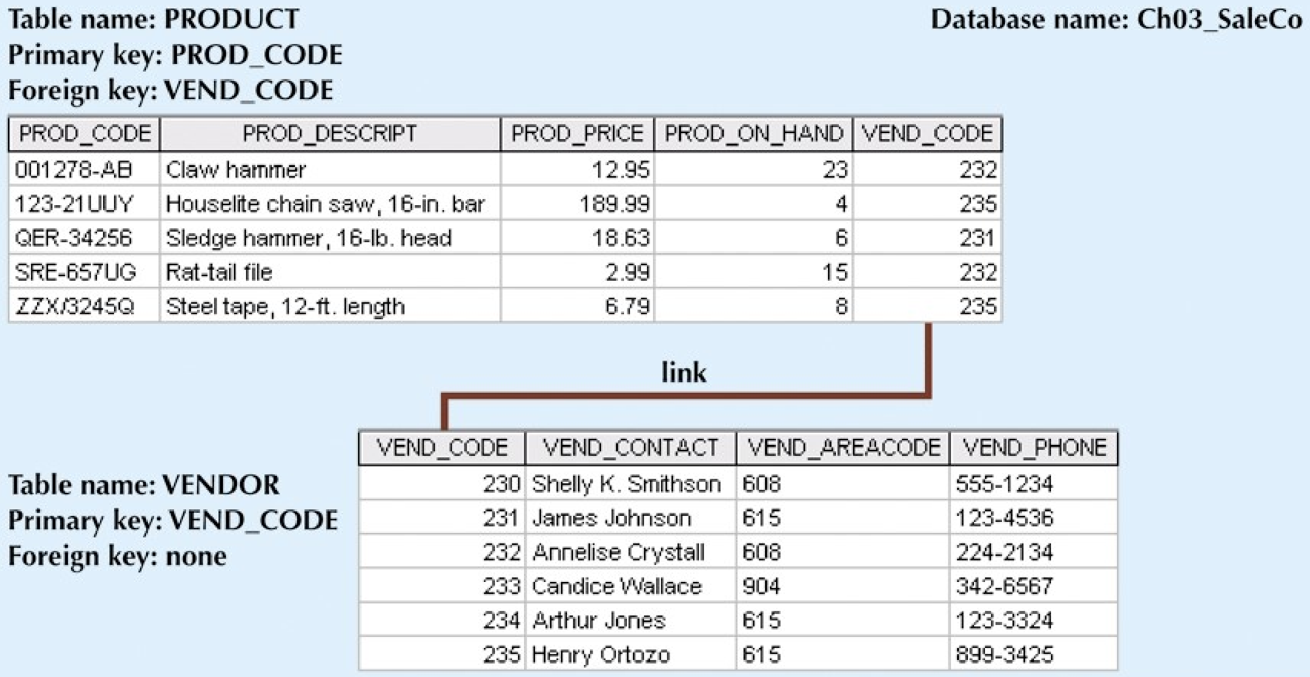


Table name: PRODUCT
Primary key: PROD_CODE
Foreign key: VEND_CODE

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

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

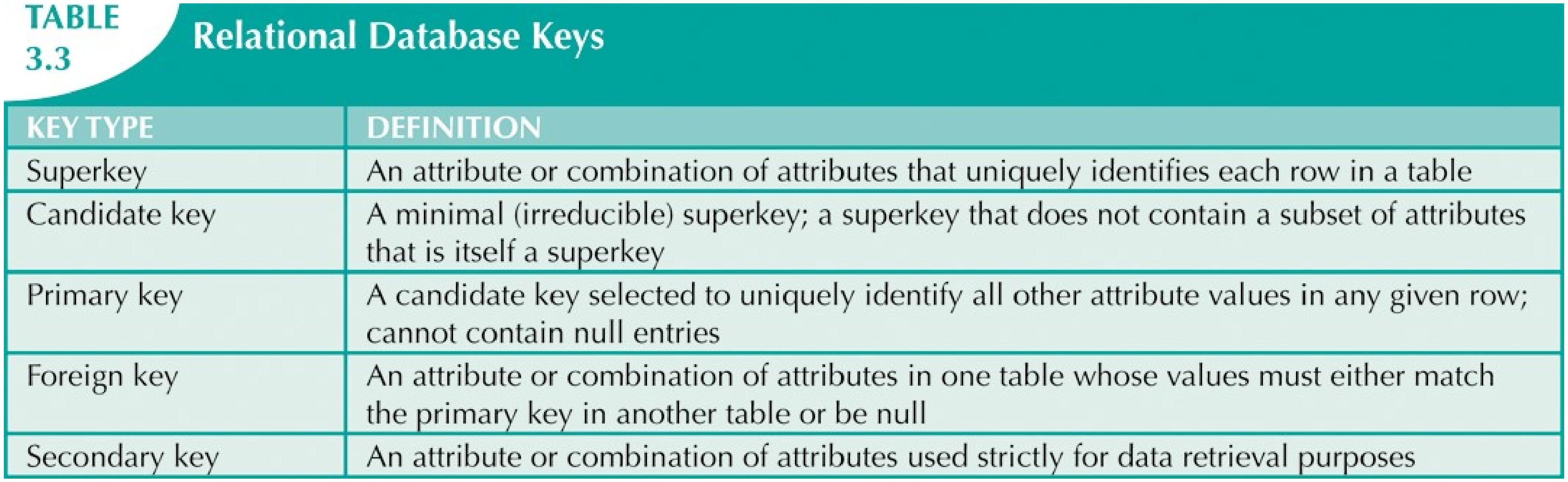


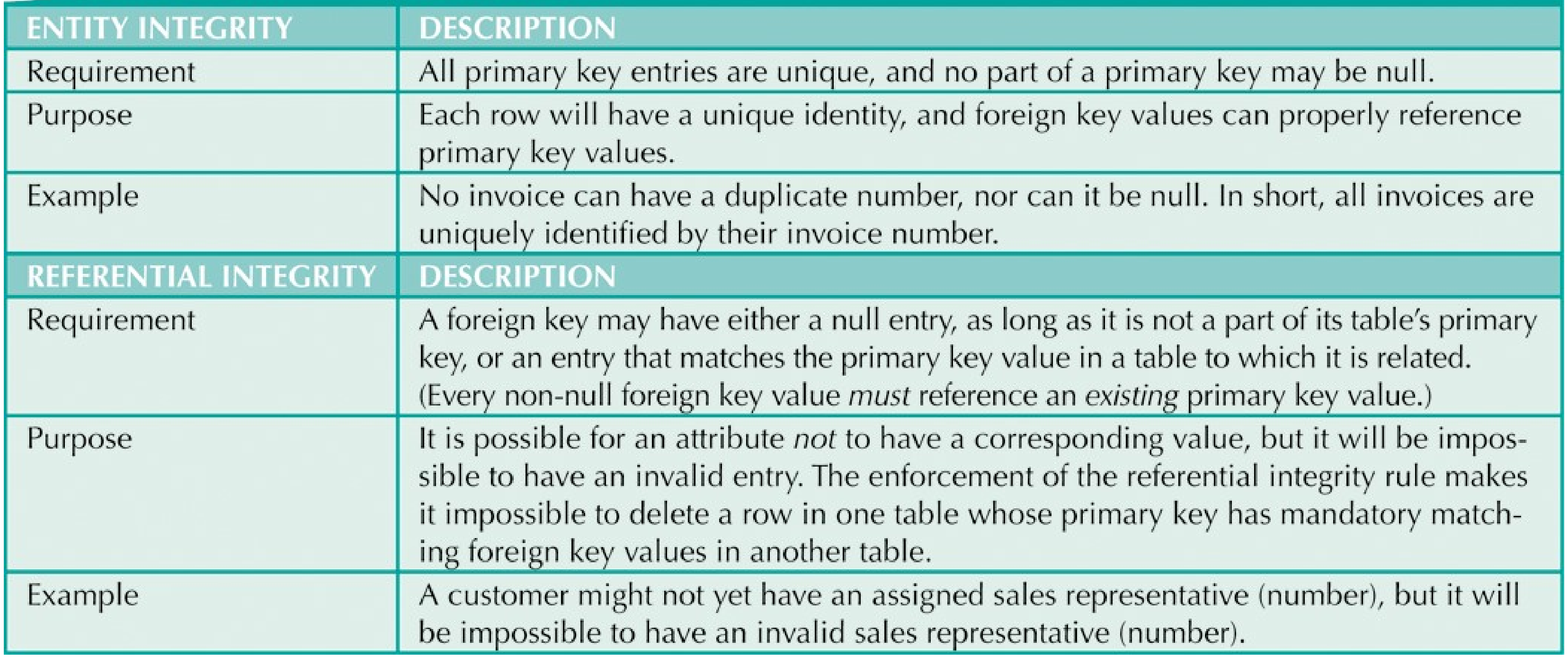
TABLE 3.3 Relational Database Keys

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

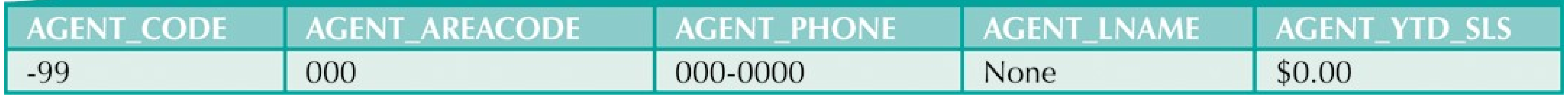
- Many RDBMs enforce integrity rules automatically
- Safer to ensure that application design conforms to entity and referential integrity rules
- Designers use flags to avoid nulls
  - Flags indicate absence of some value

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).

TABLE 3.5 A Dummy Variable Value Used as a Flag



AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
-99	000	000-0000	None	$0.00

FIGURE 3.3 An illustration of integrity rules

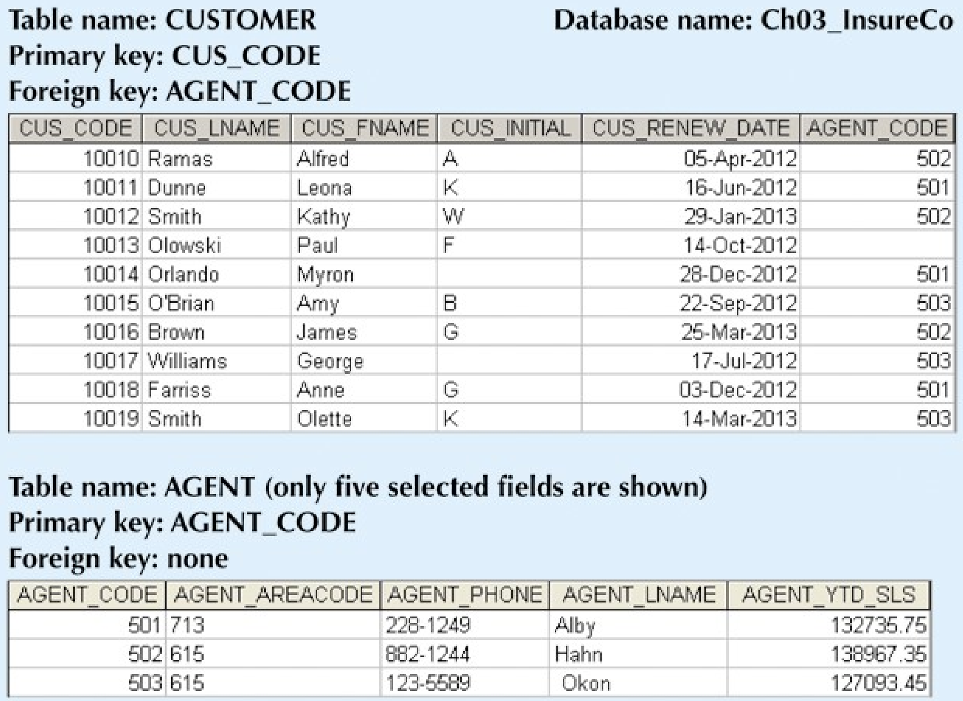


Table name: CUSTOMER **Database name: Ch03_InsureCo**
Primary key: CUS_CODE
Foreign key: AGENT_CODE

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