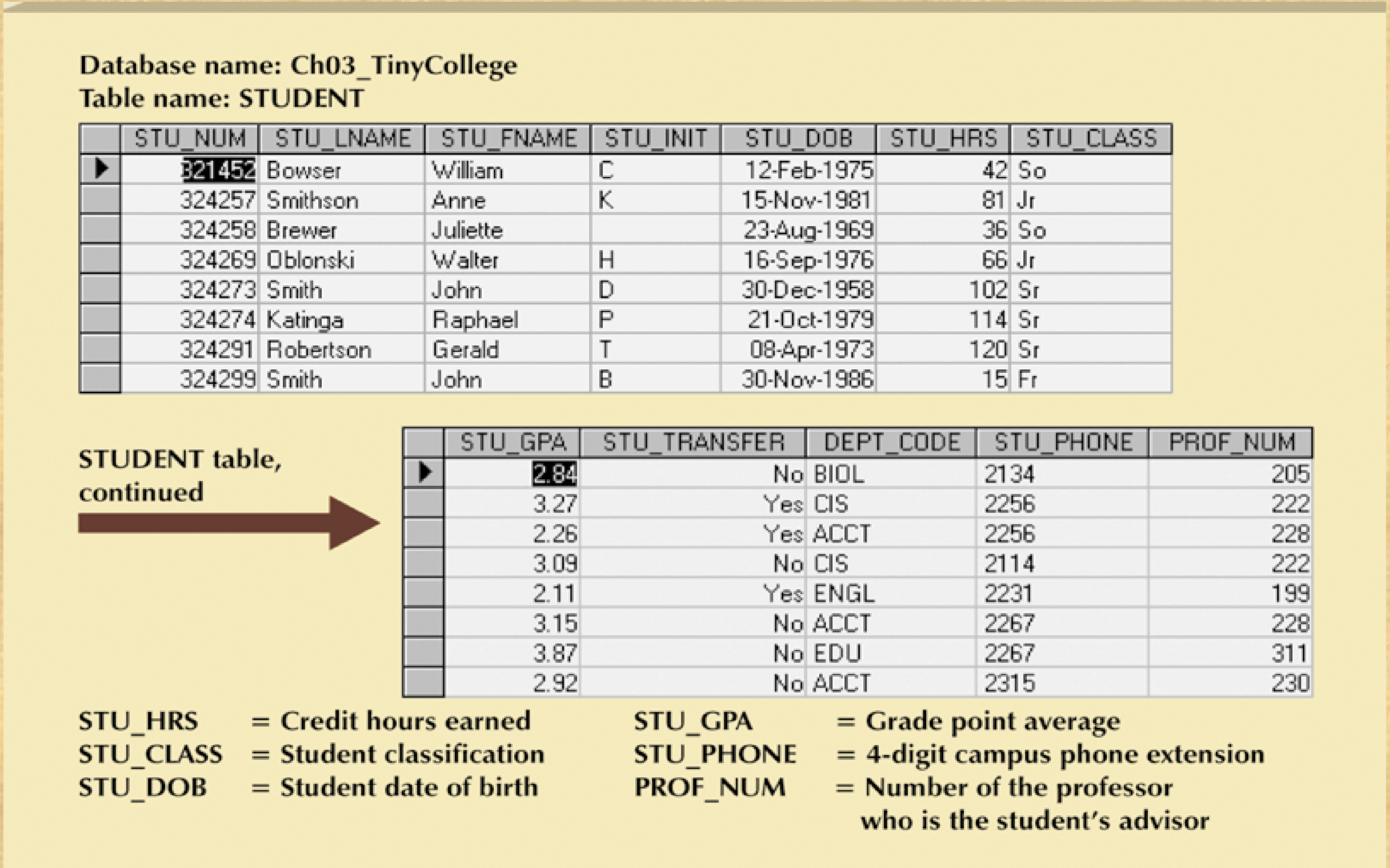
Keys

- In the relational model, keys are important because they are used to ensure that each row in a table is uniquely identifiable.
- Consists of one or more attributes that determine other attributes.
- **Primary key (PK)** is an attribute (or a combination of attributes) that uniquely identifies any given entity (row)
- Key’s role is based on determination
- If you know the value of attribute A, you can look up (determine) the value of attribute B
- STU_NUM → STU_LNAME
- STU_NUM → STU_LNAME, STU_FNAME, STU_INIT
- STU_NUM→STU_LNAME,STU_FNAME,STU_INIT,STU_DOB, STU_TRANSFER

Example Tables



Database name: Ch03_TinyCollege
Table name: STUDENT

	STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS
▶	321452	Bowser	William	C	12-Feb-1975	42	So
	324257	Smithson	Anne	K	15-Nov-1981	81	Jr
	324258	Brewer	Juliette		23-Aug-1969	36	So
	324269	Oblonski	Walter	H	16-Sep-1976	66	Jr
	324273	Smith	John	D	30-Dec-1958	102	Sr
	324274	Katinga	Raphael	P	21-Oct-1979	114	Sr
	324291	Robertson	Gerald	T	08-Apr-1973	120	Sr
	324299	Smith	John	B	30-Nov-1986	15	Fr

STUDENT table, continued

	STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
▶	2.84	No	BIOL	2134	205
	3.27	Yes	CIS	2256	222
	2.26	Yes	ACCT	2256	228
	3.09	No	CIS	2114	222
	2.11	Yes	ENGL	2231	199
	3.15	No	ACCT	2267	228
	3.87	No	EDU	2267	311
	2.92	No	ACCT	2315	230

STU_HRS	=	Credit hours earned
STU_CLASS	=	Student classification
STU_DOB	=	Student date of birth
STU_GPA	=	Grade point average
STU_PHONE	=	4-digit campus phone extension
PROF_NUM	=	Number of the professor who is the student's advisor

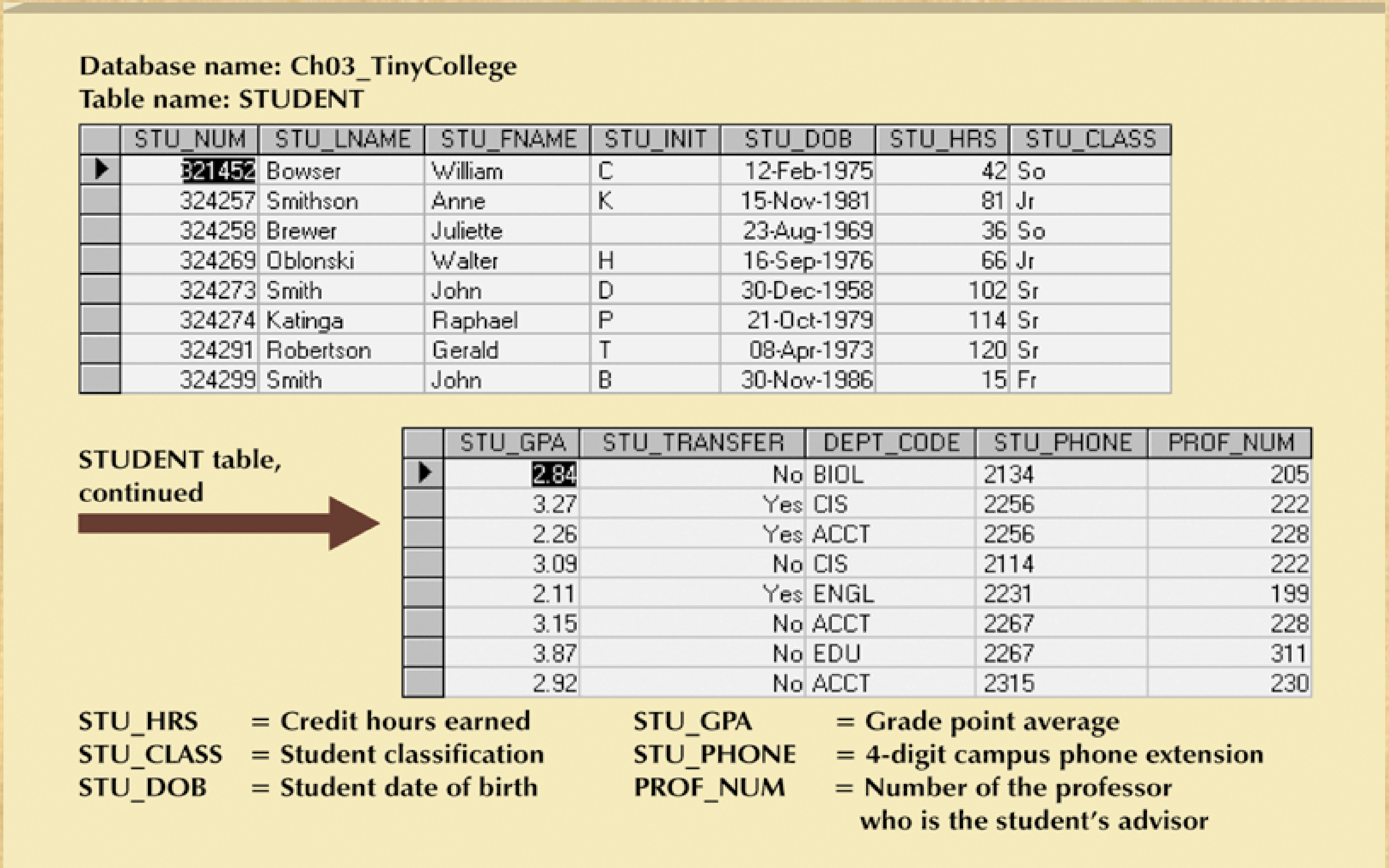
Keys (Cont'd)

Superkey

  - Any key that uniquely identifies each entity

- In the STUDENT table, the superkey could be any of the following:
- STU_NUM
- STU_NUM, STU_LNAME
- STU_NUM, STU_LNAME, STU_INIT
- In fact, STU_NUM, with or without additional attributes, can be a superkey even when the additional attributes are redundant.

Example Tables



Database name: Ch03_TinyCollege
Table name: STUDENT

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS
321452	Bowser	William	C	12-Feb-1975	42	So
324257	Smithson	Anne	K	15-Nov-1981	81	Jr
324258	Brewer	Juliette		23-Aug-1969	36	So
324269	Oblonski	Walter	H	16-Sep-1976	66	Jr
324273	Smith	John	D	30-Dec-1958	102	Sr
324274	Katinga	Raphael	P	21-Oct-1979	114	Sr
324291	Robertson	Gerald	T	08-Apr-1973	120	Sr
324299	Smith	John	B	30-Nov-1986	15	Fr

STUDENT table, continued

STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
2.84	No	BIOL	2134	205
3.27	Yes	CIS	2256	222
2.26	Yes	ACCT	2256	228
3.09	No	CIS	2114	222
2.11	Yes	ENGL	2231	199
3.15	No	ACCT	2267	228
3.87	No	EDU	2267	311
2.92	No	ACCT	2315	230

STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_DOB = Student date of birth

STU_GPA = Grade point average
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

Keys (Cont’d)

- **Candidate key**
  - A superkey without redundancies
  - A candidate key can be described as a minimal superkey.
  - STU_SSN and STU_NUM
- STU_LNAME, STU_FNAME, STU_INIT, STU_PHONE
- might also be a candidate key, as long as you discount the possibility that two students share the same last name, first name, initial, and phone number.
- **Composite key**
  - Composed of more than one attribute
- STU_LNAME, STU_FNAME, STU_INIT, STU_PHONE → STU_HRS, STU_CLASS

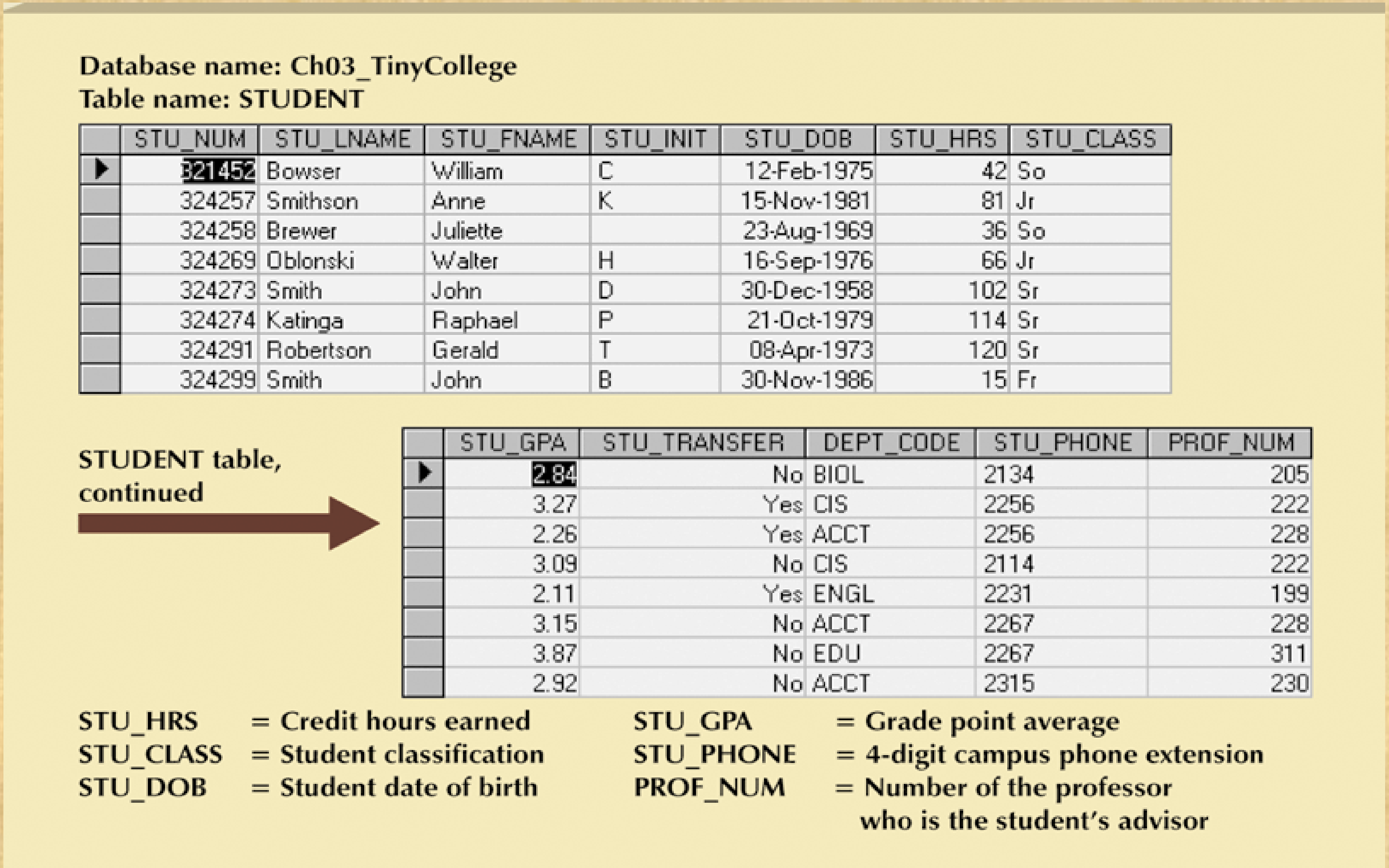
or

- STU_LNAME,STU_FNAME, STU_INIT, STU_PHONE → STU_HRS, STU_CLASS, STU_GPA

or

- STU_LNAME,STU_FNAME, STU_INIT, STU_PHONE → STU_HRS, STU_CLASS, STU_GPA, STU_DOB

Example Tables



Database name: Ch03_TinyCollege
Table name: STUDENT

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS
321452	Bowser	William	C	12-Feb-1975	42	So
324257	Smithson	Anne	K	15-Nov-1981	81	Jr
324258	Brewer	Juliette		23-Aug-1969	36	So
324269	Oblonski	Walter	H	16-Sep-1976	66	Jr
324273	Smith	John	D	30-Dec-1958	102	Sr
324274	Katinga	Raphael	P	21-Oct-1979	114	Sr
324291	Robertson	Gerald	T	08-Apr-1973	120	Sr
324299	Smith	John	B	30-Nov-1986	15	Fr

STUDENT table, continued

STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
2.84	No	BIOL	2134	205
3.27	Yes	CIS	2256	222
2.26	Yes	ACCT	2256	228
3.09	No	CIS	2114	222
2.11	Yes	ENGL	2231	199
3.15	No	ACCT	2267	228
3.87	No	EDU	2267	311
2.92	No	ACCT	2315	230

STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_DOB = Student date of birth
STU_GPA = Grade point average
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

Keys (Cont'd)

- **Foreign key (FK)**
  - An attribute whose values match primary key values in the related table
- **Secondary key**
  - Key used strictly for data retrieval purposes

Simple Relational Database

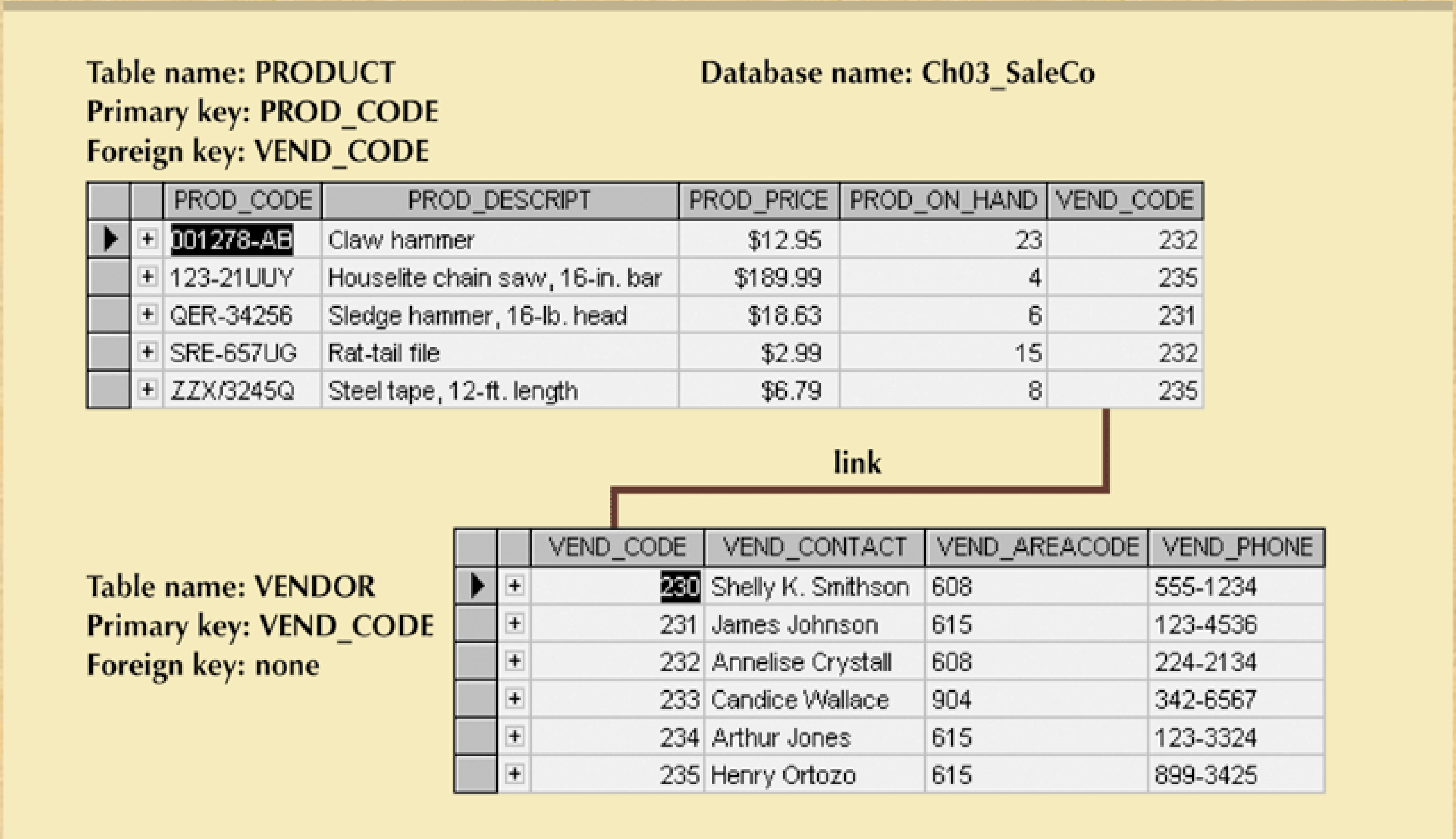


Table name: PRODUCT
Primary key: PROD_CODE
Foreign key: VEND_CODE

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

Functional dependence

- The term functional dependence can be defined most easily this way: the attribute B is functionally dependent on A if A determines B.

More precisely:

- **The attribute B is functionally dependent on the attribute A if each value in column A determines *one and only one value in column B.***

Functional dependence

- STU_PHONE is functionally dependent on STU_NUM.
  - For example, the STU_NUM value 321452 determines the STU_PHONE value 2134. On
- the other hand, STU_NUM is not functionally dependent on STU_PHONE
  - because the STU_PHONE value 2267 is associated with two STU_NUM values: 324274 and 324291.(dormitory situation)

Full Functional Dependence

- The notion of functional dependence can be further refined by specifying full functional dependence:

  If the attribute (B) is functionally dependent on a composite key (A) but not on any subset of that composite key, the attribute (B) is fully functionally dependent on (A).

Full Functional Dependence

- STU_LNAME, STU_FNAME, STU_INIT, STU_PHONE → STU_HRS, STU_CLASS

or

- STU_LNAME,STU_FNAME, STU_INIT, STU_PHONE → STU_HRS, STU_CLASS, STU_GPA

or

- STU_LNAME,STU_FNAME, STU_INIT, STU_PHONE → STU_HRS, STU_CLASS, STU_GPA, STU_DOB

Entity Integrity and Null value:

- Within a table, each primary key value must be unique to ensure that each row is uniquely identified by the primary key. In that case, the table is said to exhibit entity integrity. To maintain entity integrity, a null (that is, no data entry at all) is not permitted in the primary key.
- A null is no value at all. It does not mean a zero or a space. A null is created when you press the Enter key or the Tab key to move to the next entry without making a prior entry of any kind. Pressing the Spacebar creates a blank (or a space).
  - There are rare cases in which nulls cannot be reasonably avoided when you are working with nonkey attributes. For example, one of an EMPLOYEE table’s attributes is likely to be the EMP_INITIAL. However, some employees do not have a middle initial. Therefore, some of the EMP_INITIAL values may be null.

Null value

- In any case, even if nulls cannot always be avoided, they must be used sparingly. In fact, the existence of nulls in a table is often an indication of poor database design.
- Nulls, if used improperly, can create problems because they have many different meanings. For example, a null can represent:
  - An unknown attribute value.
  - A known, but missing, attribute value.
  - A “not applicable” condition.

Controlled Redundancy

- Makes the relational database work
- Tables within the database share common attributes that enable us to link tables together
- Multiple occurrences of values in a table are not redundant when they are *required* to make the relationship work
- Redundancy is *unnecessary* duplication of data

- For example, note that the PRODUCT and VENDOR tables in Figure share a common attribute named VEND_CODE. And note that the PRODUCT table's VEND_CODE value 232 occurs more than once, as does the VEND_CODE value 235. Because the PRODUCT table is related to the VENDOR table through these VEND_CODE values, the multiple occurrence of the values is required to make the 1:M relationship between VENDOR and PRODUCT work.
- Each VEND_CODE value in the VENDOR table is unique—the VENDOR is the "1" side in the VENDOR-PRODUCT relationship. But any given VEND_CODE value from the VENDOR table may occur more than once in the PRODUCT table, thus providing evidence that PRODUCT is the "M" side of the VENDOR-PRODUCT relationship

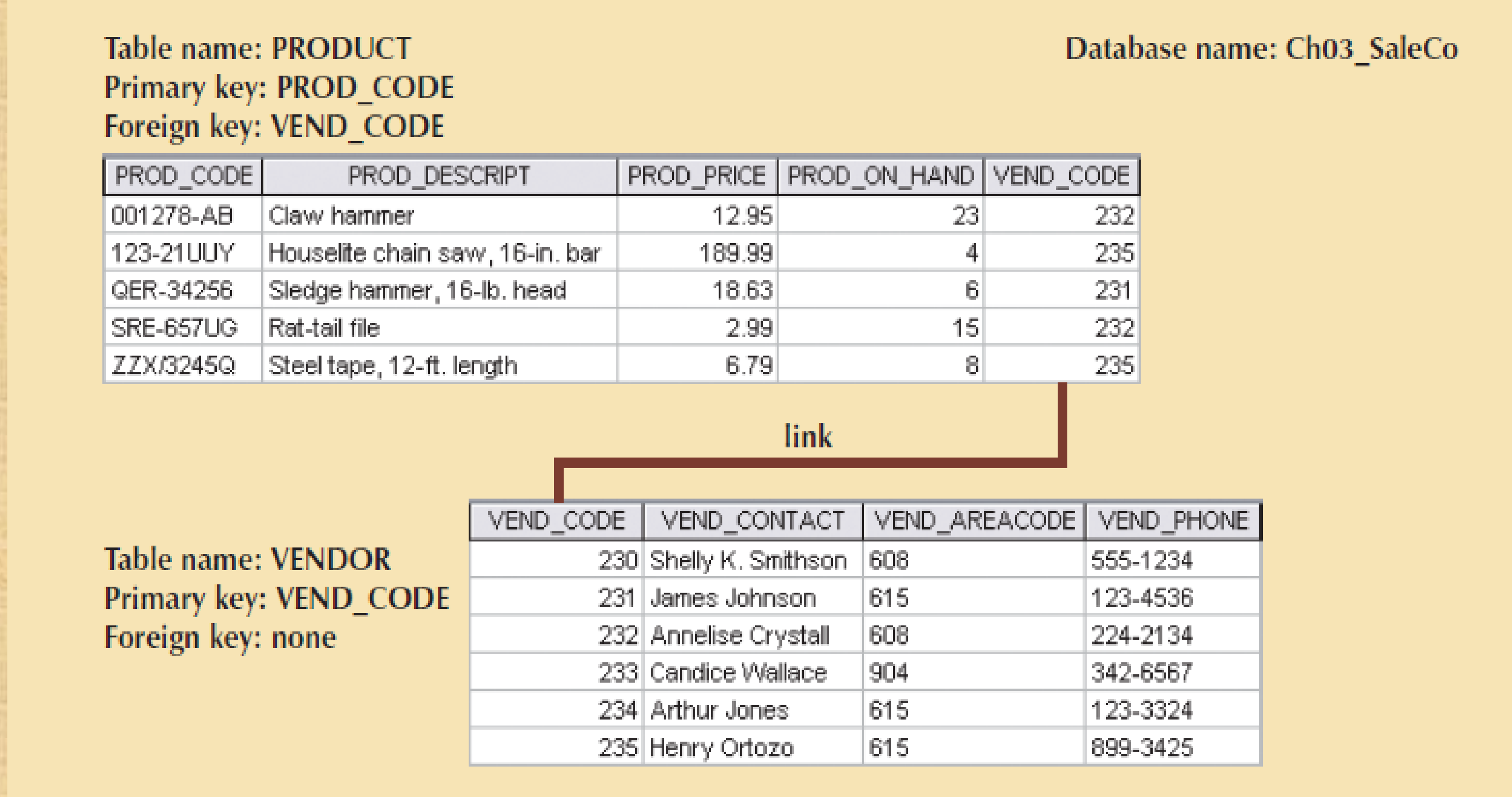


Table name: PRODUCT
Primary key: PROD_CODE
Foreign key: VEND_CODE

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

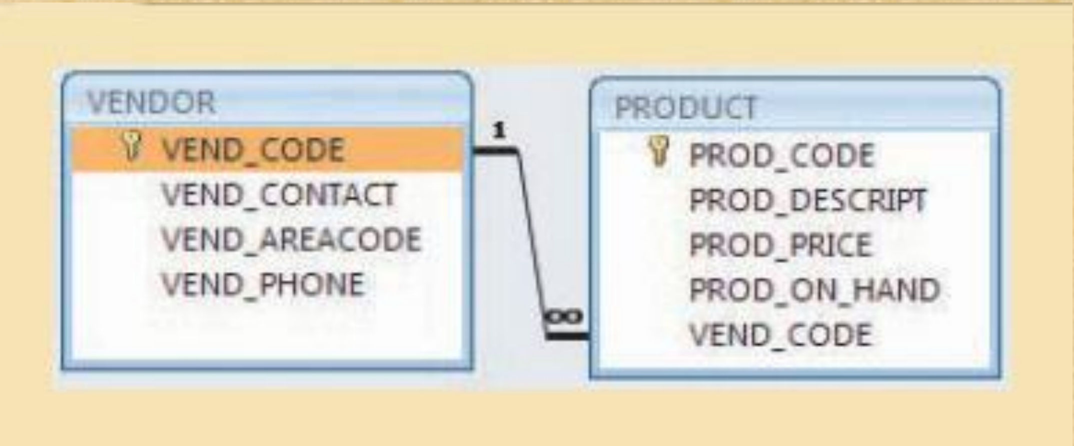
VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Relational Schema

- A relational schema is a textual representation of the database tables where each table is listed by its name followed by the list of its attributes in parentheses.
- For Examples:

VENDOR (VEND_CODE, VEND_CONTACT, VEND_AREACODE, VEND_PHONE)
PRODUCT (PROD_CODE, PROD_DESCRIPT, PROD_PRICE, PROD_ON_HAND, VEND_CODE)

Represented By The Relational Diagram



Foreign Key (FK) and Referential Integrity

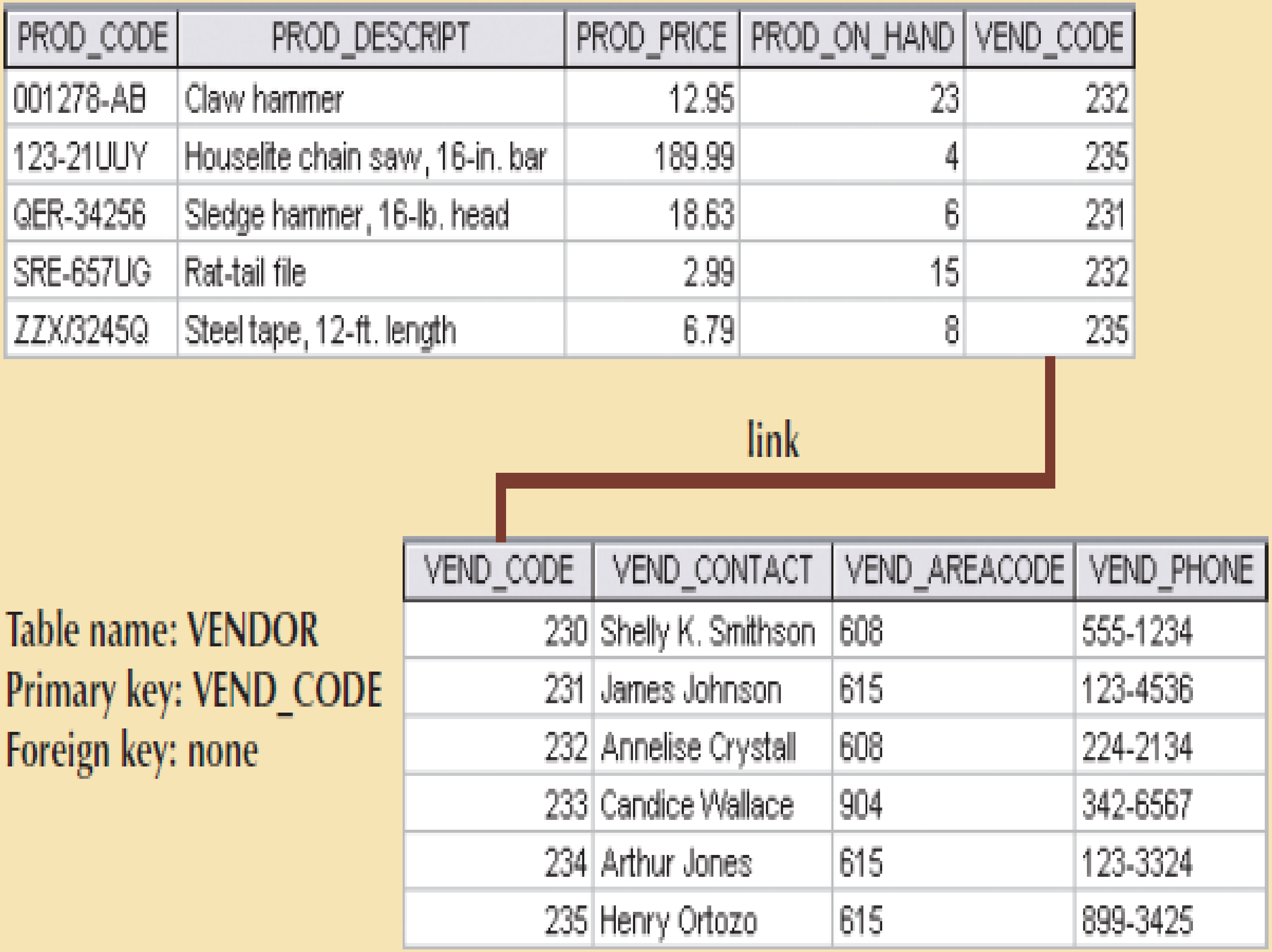
- A foreign key (FK) is an attribute whose values match the primary key values in the related table. For example, in Figure, the VEND_CODE is the primary key in the VENDOR table, and it occurs as a foreign key in the PRODUCT table. Because the VENDOR table is not linked to a third table, the VENDOR table shown in Figure, does not contain a foreign key.
- If the foreign key contains either matching values or nulls, the table that makes use of that foreign key is said to exhibit referential integrity. In other words, referential integrity means that if the foreign key contains a value, that value refers to an existing valid tuple (row) in another relation. Note that referential integrity is maintained between the PRODUCT and VENDOR tables shown in Figure

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

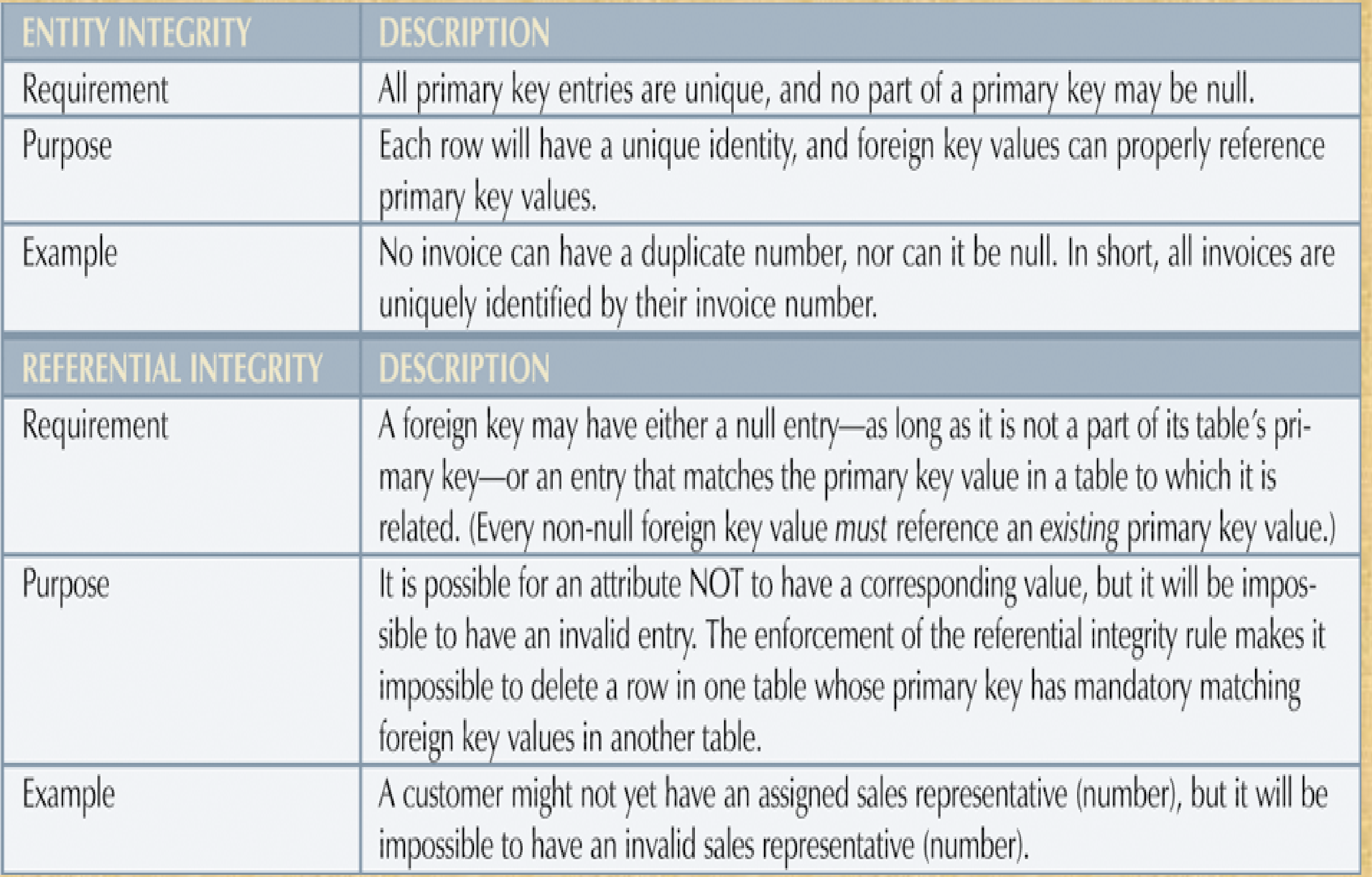
Table name: VENDOR

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

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).

Integrity Rules (cont'd)

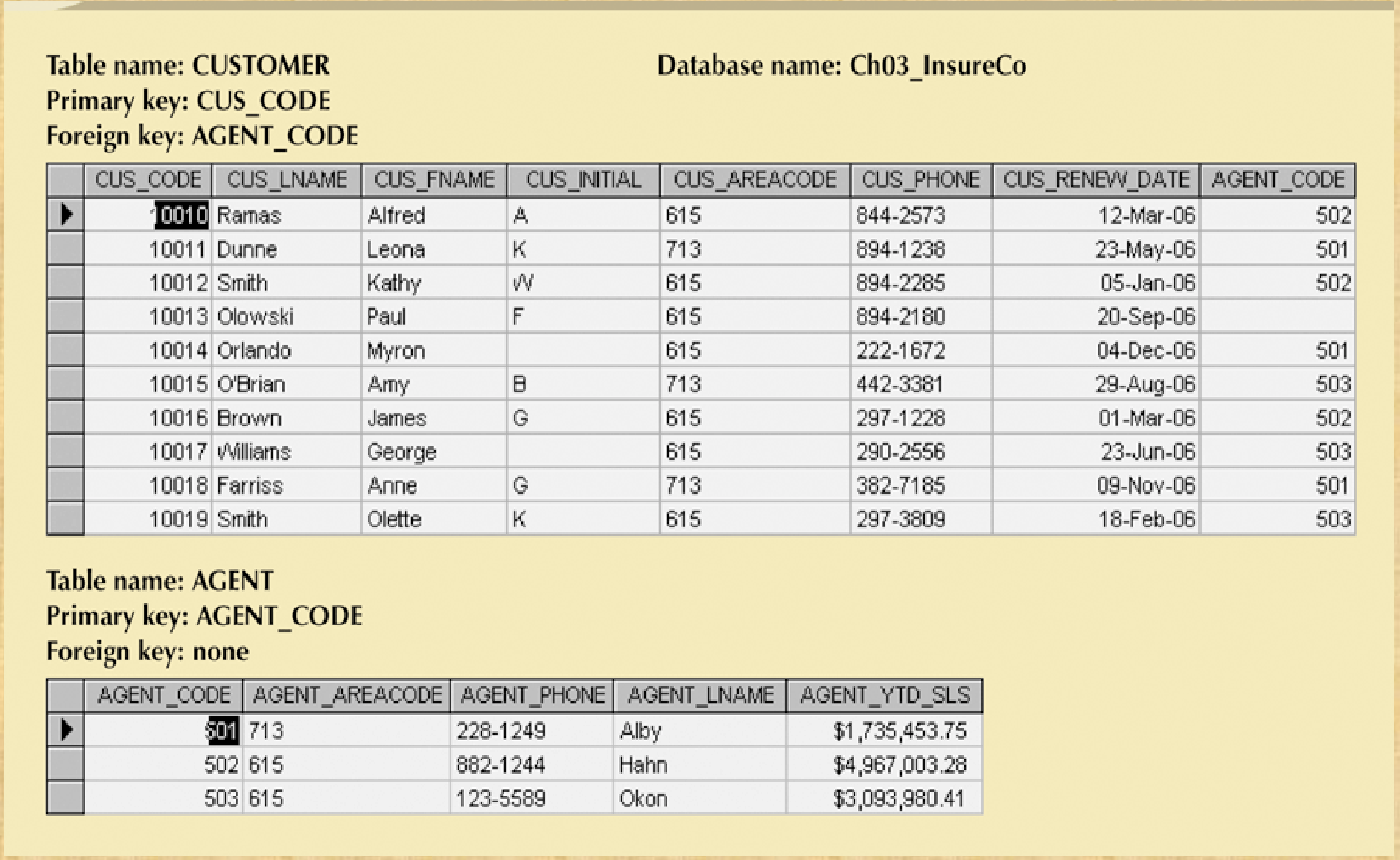


Table name: CUSTOMER

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_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	12-Mar-06	502
10011	Dunne	Leona	K	713	894-1238	23-May-06	501
10012	Smith	Kathy	W	615	894-2285	05-Jan-06	502
10013	Olowski	Paul	F	615	894-2180	20-Sep-06	
10014	Orlando	Myron		615	222-1672	04-Dec-06	501
10015	O'Brian	Amy	B	713	442-3381	29-Aug-06	503
10016	Brown	James	G	615	297-1228	01-Mar-06	502
10017	Williams	George		615	290-2556	23-Jun-06	503
10018	Farriss	Anne	G	713	382-7185	09-Nov-06	501
10019	Smith	Olette	K	615	297-3809	18-Feb-06	503

Table name: AGENT

Primary key: AGENT_CODE

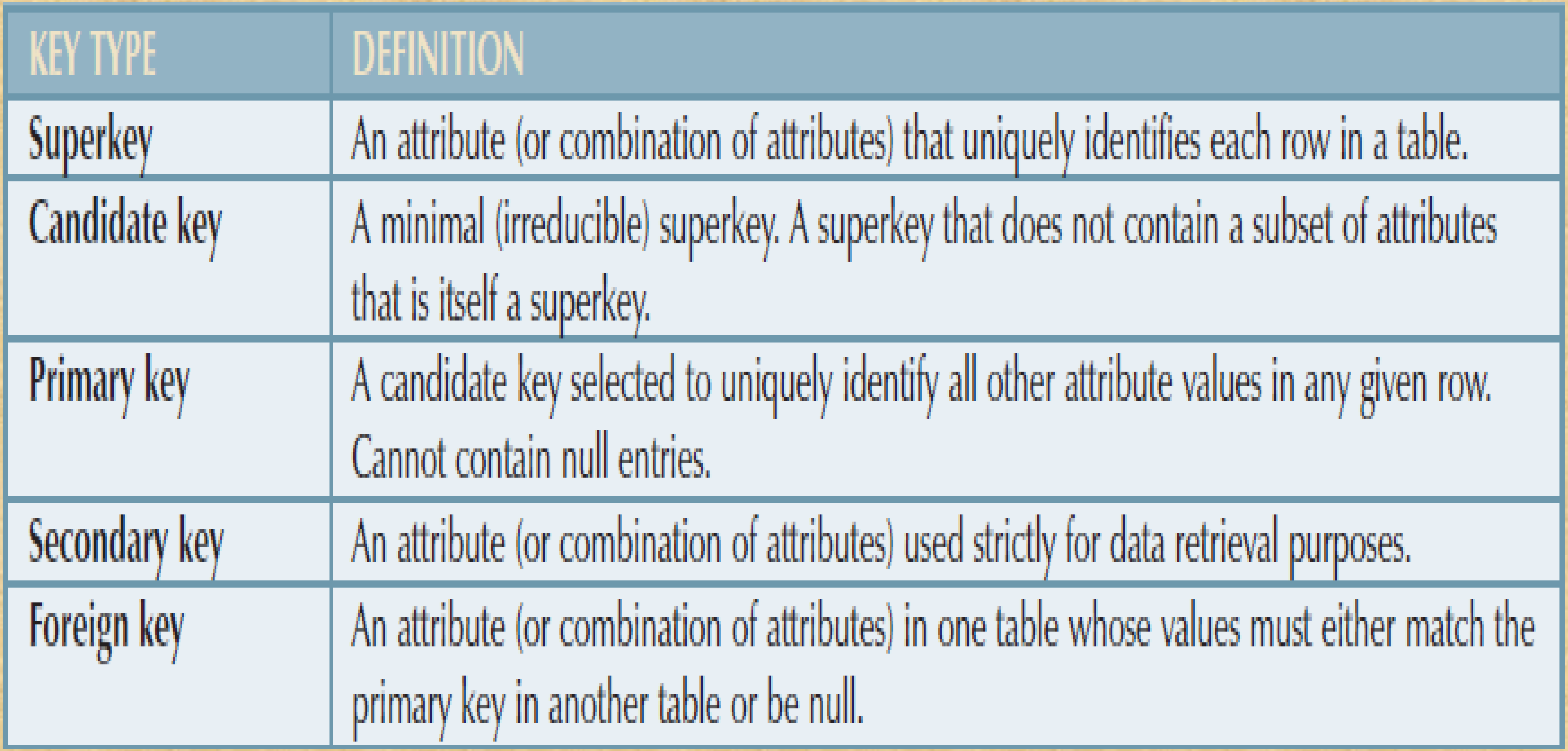
Foreign key: none

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
501	713	228-1249	Alby	$1,735,453.75
502	615	882-1244	Hahn	$4,967,003.28
503	615	123-5589	Okon	$3,093,980.41

Entity integrity & Referential integrity

- **Entity integrity.** The CUSTOMER table's primary key is CUS_CODE. The CUSTOMER primary key column has no null entries, and all entries are unique. Similarly, the AGENT table's primary key is AGENT_CODE, and this primary key column also is free of null entries.
- **Referential integrity.** The CUSTOMER table contains a foreign key AGENT_CODE, which links entries in the CUSTOMER table to the AGENT table. The CUS_CODE row that is identified by the (primary key) number 10013 contains a null entry in its AGENT_CODE foreign key because Mr. Paul F. Olowski does not yet have a sales representative assigned to him. The remaining AGENT_CODE entries in the CUSTOMER table all match the AGENT_CODE entries in the AGENT table.

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.

Flags

- To avoid nulls, some designers use special codes, known as flags, to indicate the absence of some value.
- The code -99 could be used as the AGENT_CODE entry of the fourth row of the CUSTOMER table to indicate that customer Paul Olowski does not yet have an agent assigned to him.

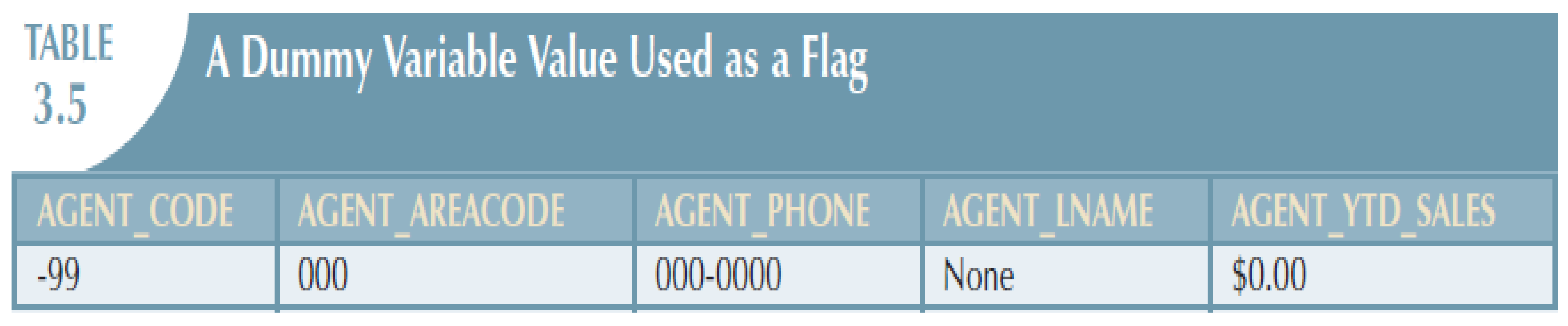


TABLE 3.5 A Dummy Variable Value Used as a Flag

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

Other Integrity Rules

- Other integrity rules that can be enforced in the relational model are the *NOT NULL and UNIQUE constraints.*
- **NOT NULL** constraint can be placed on a column to ensure that every row in the table has a value for that column.
- The **UNIQUE** constraint is a restriction placed on a column to ensure that no duplicate values exist for that column.

End of Lecture

Any Questions... ?