Database Security

CT069-3-3

A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Relational Model

Learning Outcomes

At the end of this module, YOU should be able to :

- Create a relational model in 3NF
- Convert an ER model into 3NF Relational Model
- Create data dictionary to complete your design

Key Terms you must be able to use

If you have mastered this topic, you should be able to use the following terms correctly :

- Relations
- Tuples , Columns
- Primary Key, Foreign Key
- Normalization and 3NF

Topic & Structure of The Lesson

- Relational Model
- Normalization

Relational Model

- Codd proposed the relational data model in 1970.
- At that time, most database systems were based on one of two older data models (the hierarchical model and the network model); the relational model revolutionized the database field and largely supplanted these earlier models.
- The relational model is by far the dominant data model
- Represents database as a collection of relations

Relation

- The main construct for representing data in the relational model is a ‘relation’.
- A relation consists of a set of attributes
- Example:
  - Book(BookID, Name, Price)
  - Relation name : Book
  - Attributes: BookID, Name, Price

Relational Table

- Relational table is an **instance of a relation** which consists of attributes and data for the attributes
- The relational table is defined to be a set of unique tuples or rows. No two rows are identical.

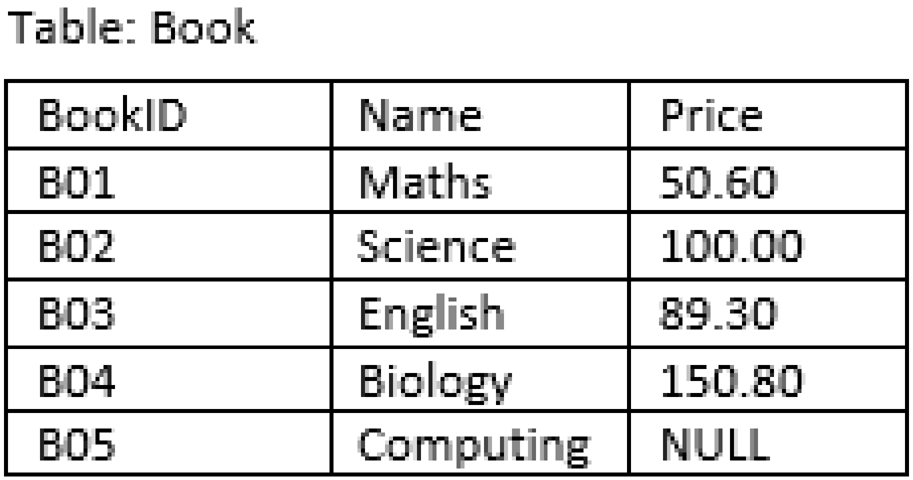


Table: Book

BookID	Name	Price
B01	Maths	50.60
B02	Science	100.00
B03	English	89.30
B04	Biology	150.80
B05	Computing	NULL

tuples

Relational Tables and Their Characteristics

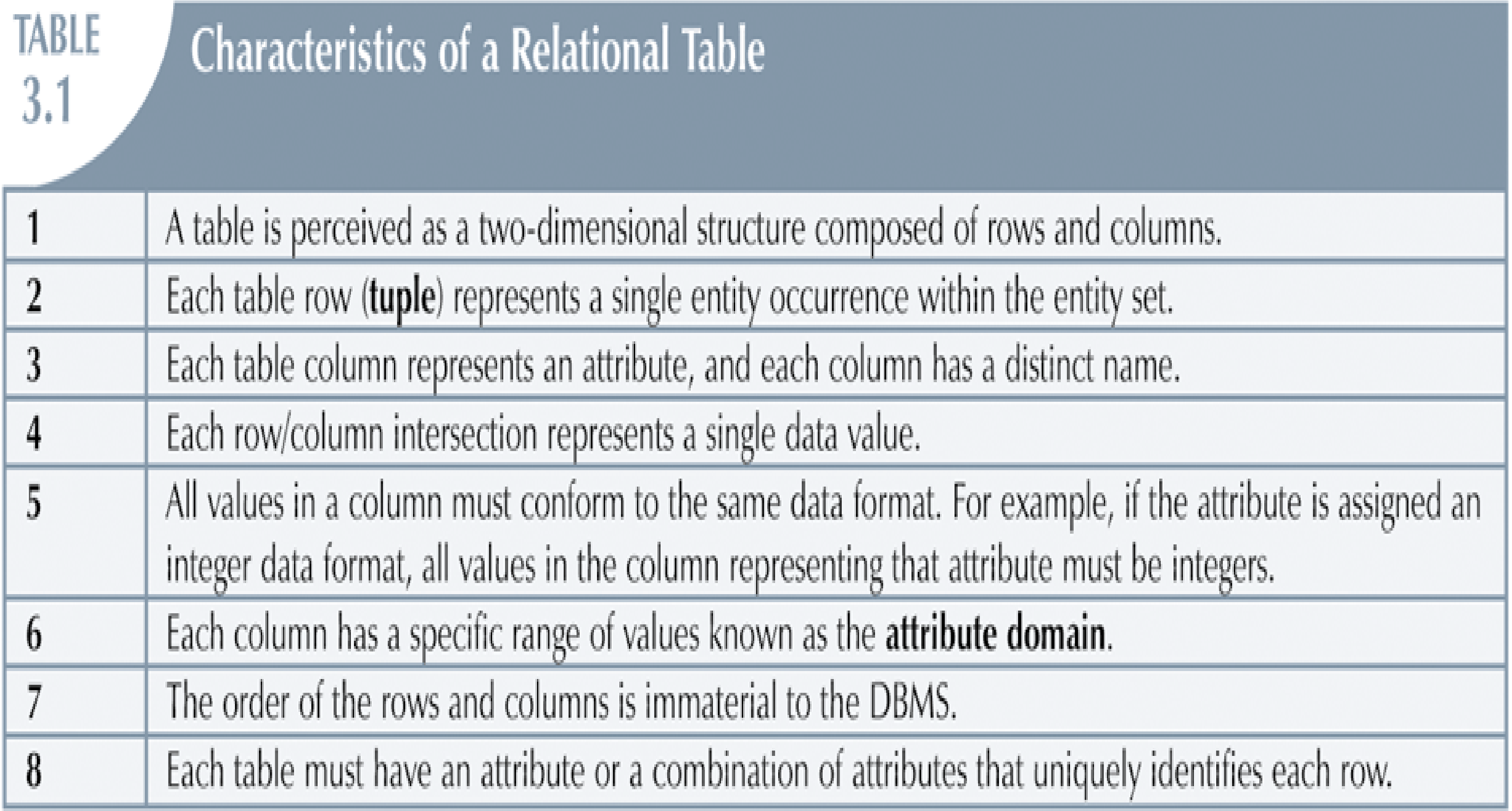


TABLE 3.1 Characteristics of a Relational Table

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. For example, if the attribute is assigned an integer data format, all values in the column representing that attribute must be integers.
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.

Degree & Cardinality

- **Degree:** The total number of attributes in the table
- **Cardinality:** Total number of rows present in the table

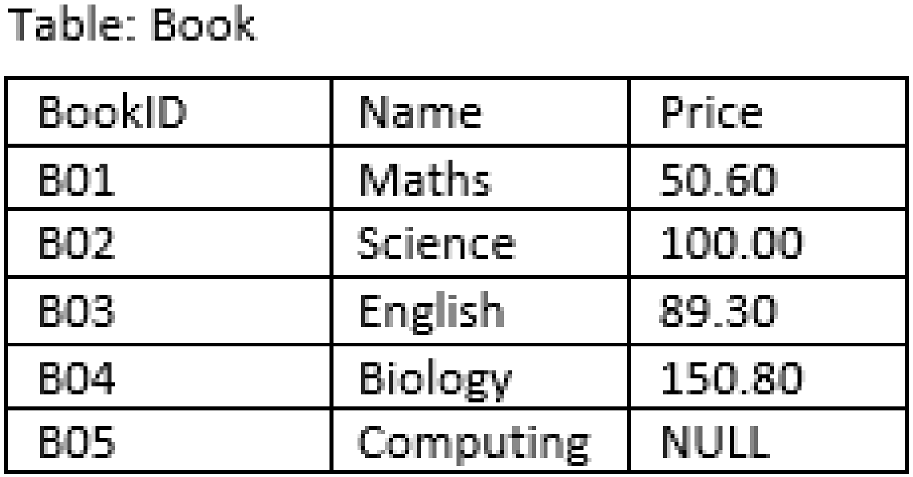


Table: Book

BookID	Name	Price
B01	Maths	50.60
B02	Science	100.00
B03	English	89.30
B04	Biology	150.80
B05	Computing	NULL

Degree = 3

Cardinality = 5

Keys

- A key is an attribute or set of attributes that determine other attributes
- 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

Student ID	Name	Gender	Passport #	Address	Mobile
S01	Ali	Male	M20090087	Sri Petaling	010-202-3945
S02	Nancy	Female	P300200	Bukit Jalil	NULL
S03	John	Male	I200900	Sri Petaling	018-400-9367
S04	Nancy	Female	J402121	Japan	012-550-1234
S05	Ram	Male	S40505000	Singapore	013-606-7899

Why we need Keys ?

- It helps you to identify a row (tuple) in a relation (table)
- It also helps you to find the relationship between two tables
- Enforce **identity** and **integrity** in the relationship.

Types of Keys

- **Primary key**
- **Foreign key**

Primary Key

- An attribute (or a combination of attributes) that uniquely identifies any given entity (row)
- Enforces Entity Integrity Rule
  - **Value for primary key cannot be null and/or repeating**

Foreign Key

- An attribute whose values match primary key values in the related table
- It creates a relationship between two tables.
- The purpose of a foreign key is to maintain data integrity and allow navigation between two different instances of an entity.
- It acts as a cross-reference between two tables as it references the primary key of another table.

Foreign Key

- **Referential** integrity rules
  1. **You can't delete a record from a primary table if matching records exist in a related table.**
  2. **You can't change a primary key value in the primary table if that record has related records.**
  3. **You can't enter a value in the foreign key field of the related table that value doesn't exist in the primary key of the primary table.**
  4. **However, you can enter a Null value in the foreign key, specifying that the records are unrelated.**

Integrity Rules

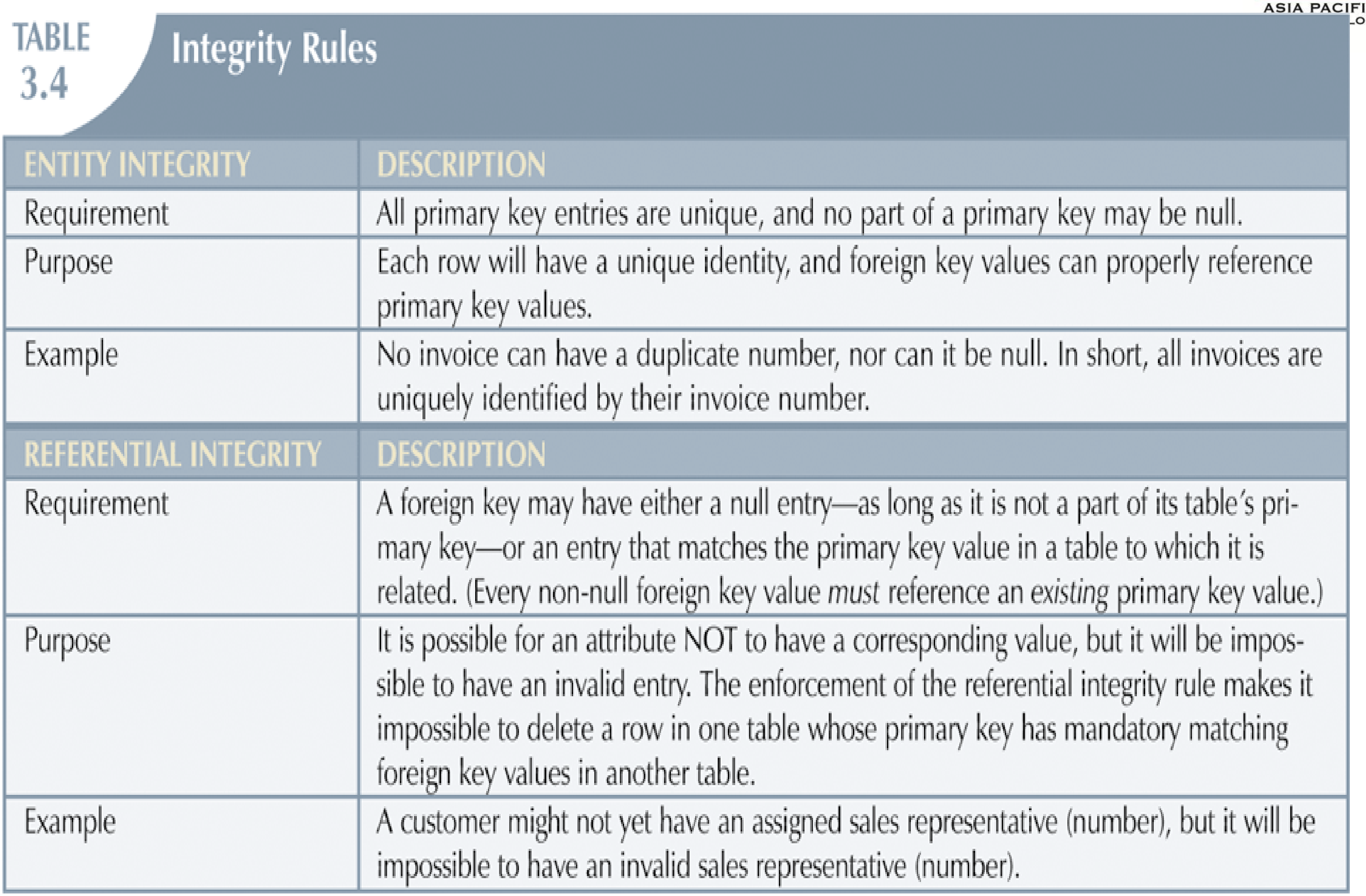


TABLE 3.4 Integrity Rules

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).

How to select Primary Key

- Select a attribute that does not contain NULL
- Select a attribute that is unique and does not repeat
- Make sure that Primary Key does not keep changing

Null

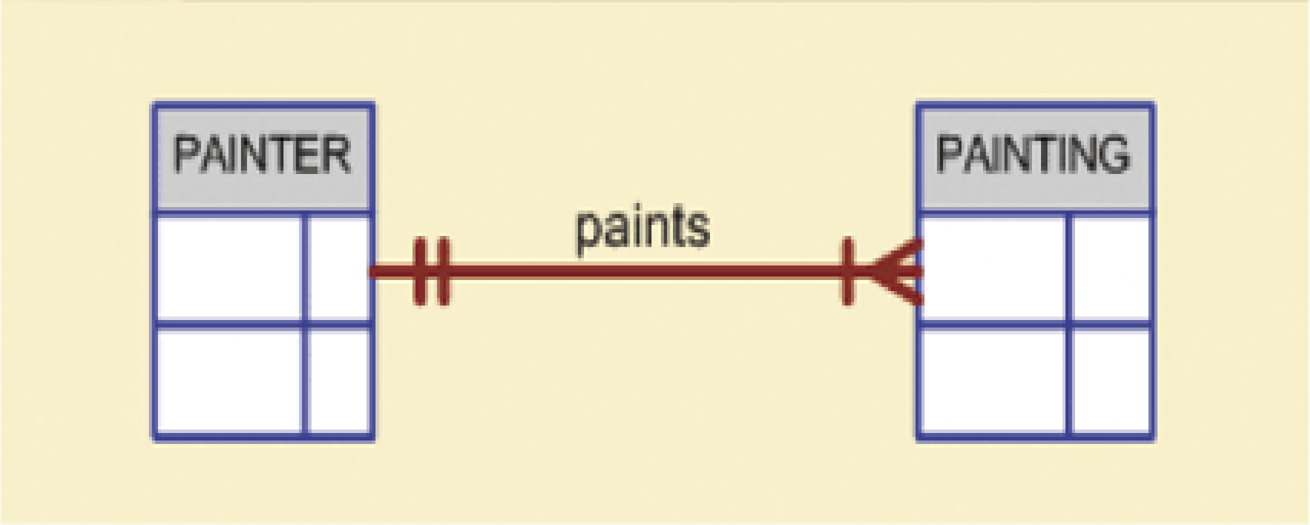
- Nulls:
  - Means absence of a value
  - Not permitted in primary key
  - Can represent
    - An unknown attribute value
    - A “not applicable” condition

The 1:M Relationship

- Relational database norm
- Found in any database environment
- 1 instance of entity A can have multiple instances of entity B

The 1:M Relationship (continued)

FIGURE 3.18 The 1:M relationship between PAINTER and PAINTING



The 1:M Relationship (continued)

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

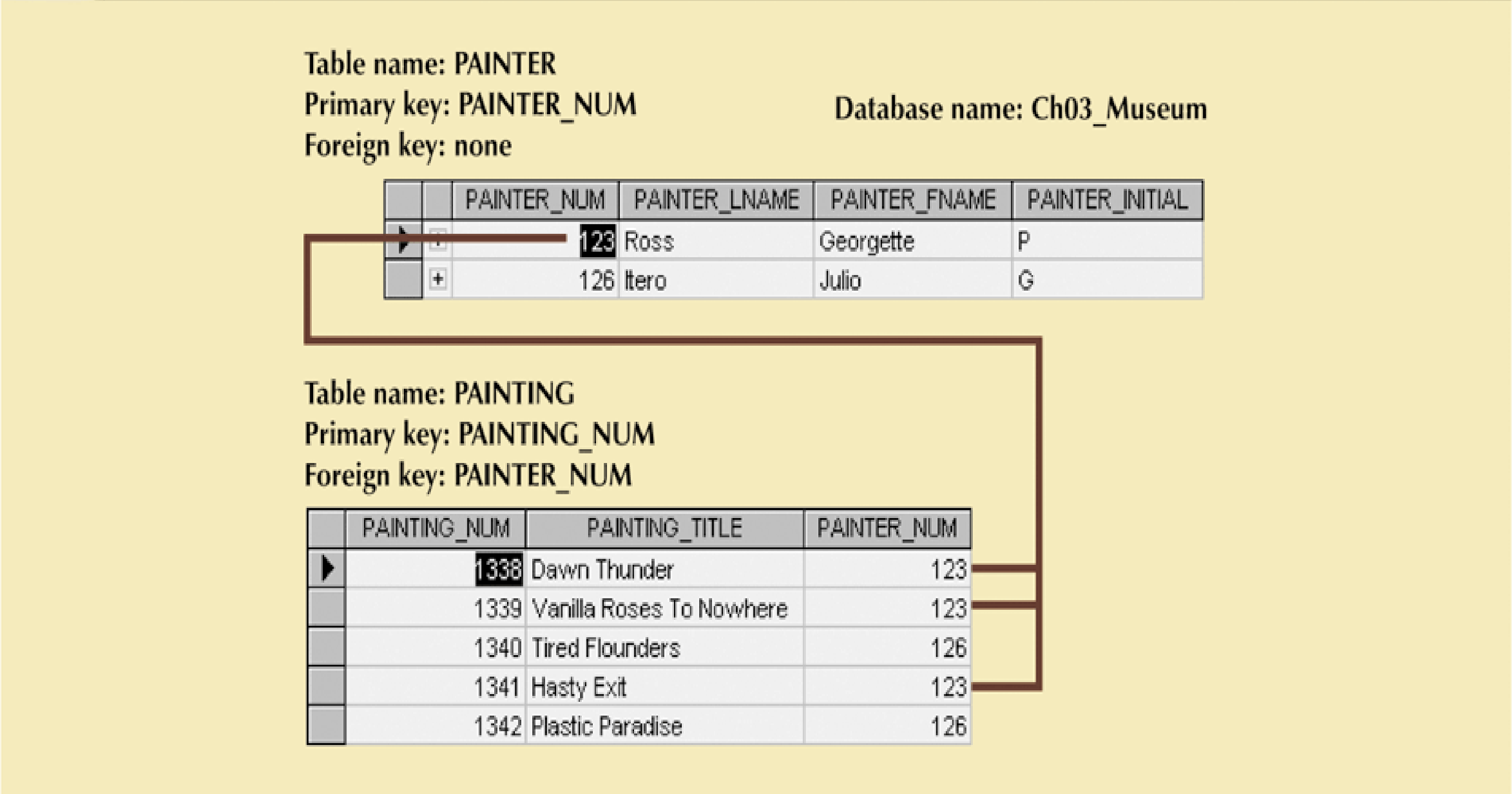


Table name: PAINTER
Primary key: PAINTER_NUM
Foreign key: none

Database name: Ch03_Museum

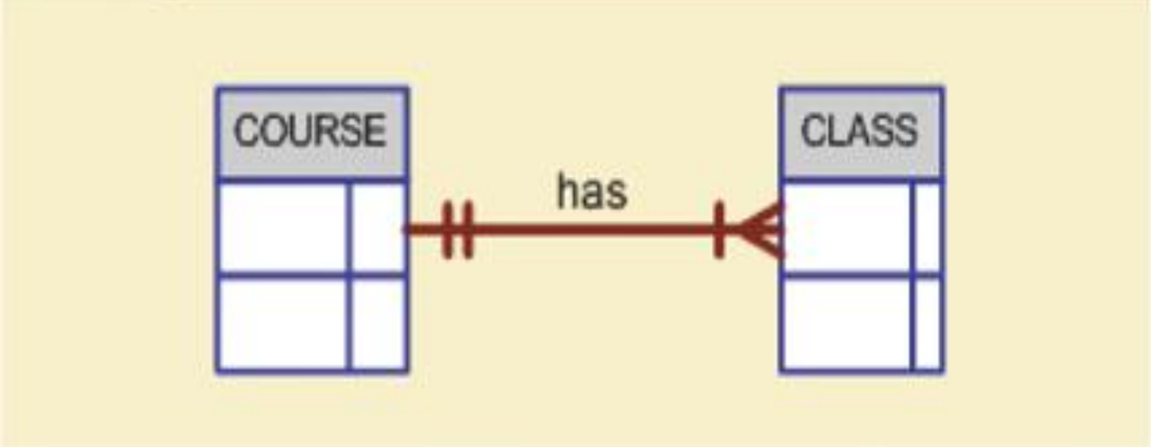
PAINTER_NUM	PAINTER_LNAME	PAINTER_FNAME	PAINTER_INITIAL
123	Ross	Georgette	P
126	Itero	Julio	G

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:M Relationship (continued)

FIGURE 3.20 The 1:M relationship between COURSE and CLASS



The 1:M Relationship (continued)

FIGURE 3.21 The implemented 1:M relationship between COURSE and CLASS

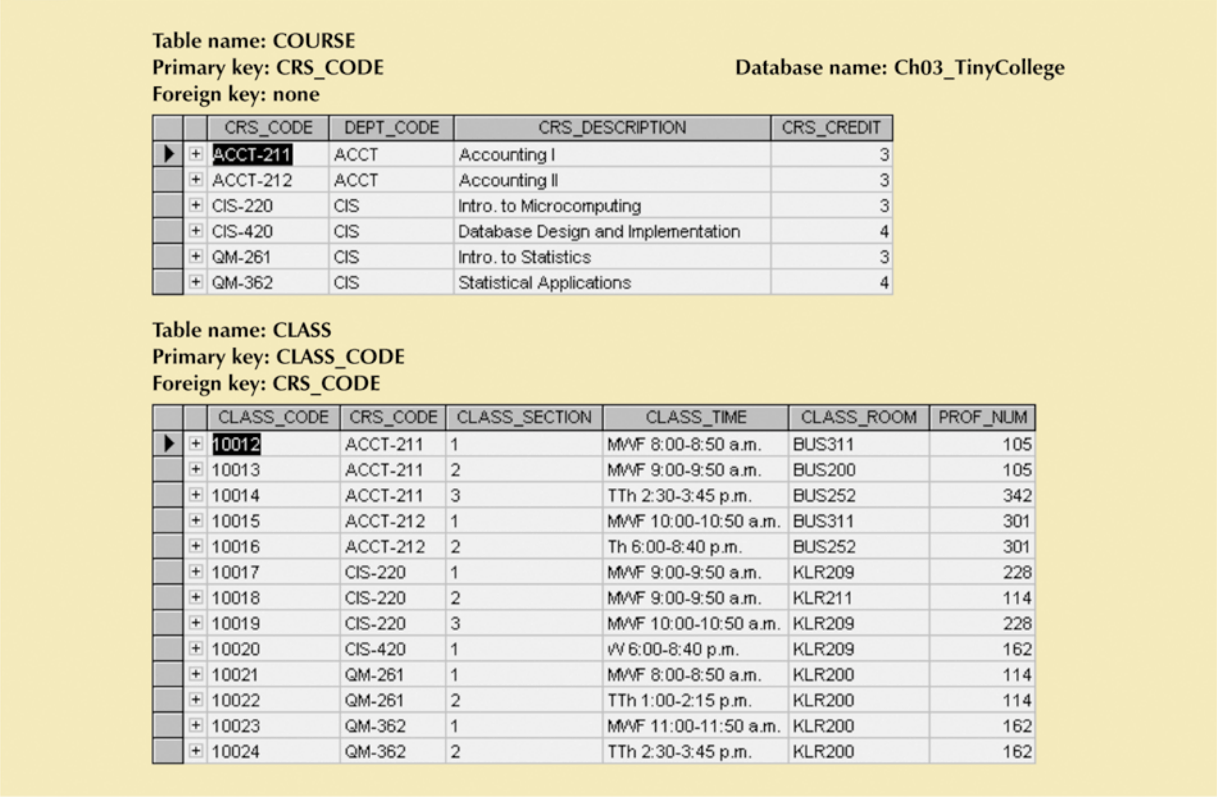


Table name: COURSE
Primary key: CRS_CODE
Foreign key: none

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

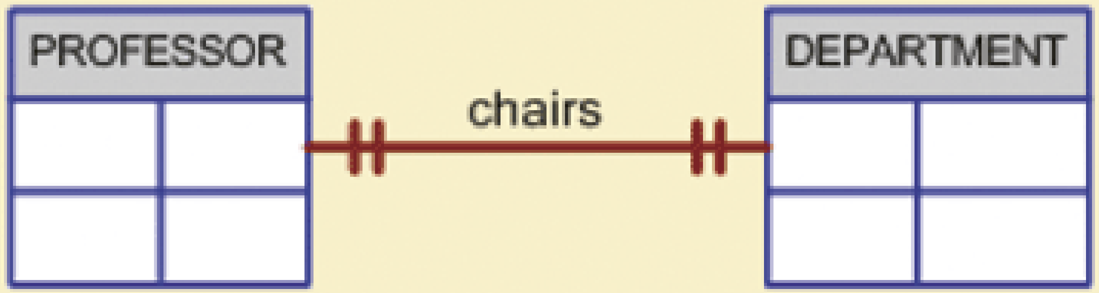
		CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
▶	+	10012	ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
	+	10013	ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
	+	10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
	+	10015	ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
	+	10016	ACCT-212	2	Th 6:00-8:40 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 instance can be related to only one other entity instance, and vice versa

The 1:1 Relationship (continued)

FIGURE 3.22 The 1:1 relationship between PROFESSOR and DEPARTMENT



The 1:1 Relationship (continued)

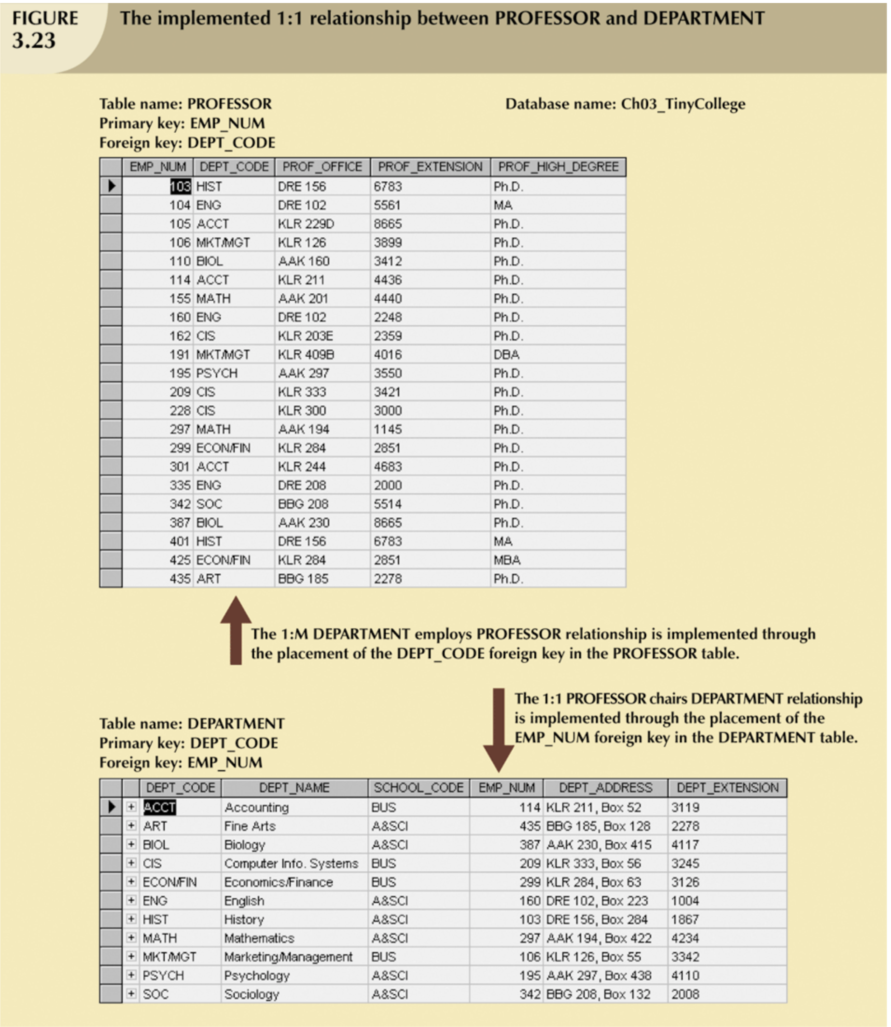


FIGURE 3.23 The implemented 1:1 relationship between PROFESSOR and DEPARTMENT

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.

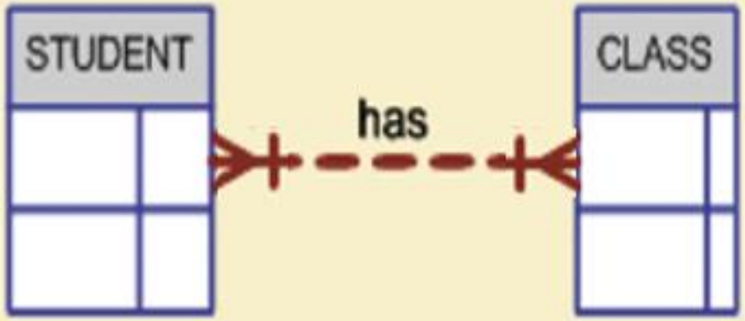
The 1:1 PROFESSOR chairs DEPARTMENT relationship is implemented through the placement of the EMP_NUM foreign key in the DEPARTMENT table.

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

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



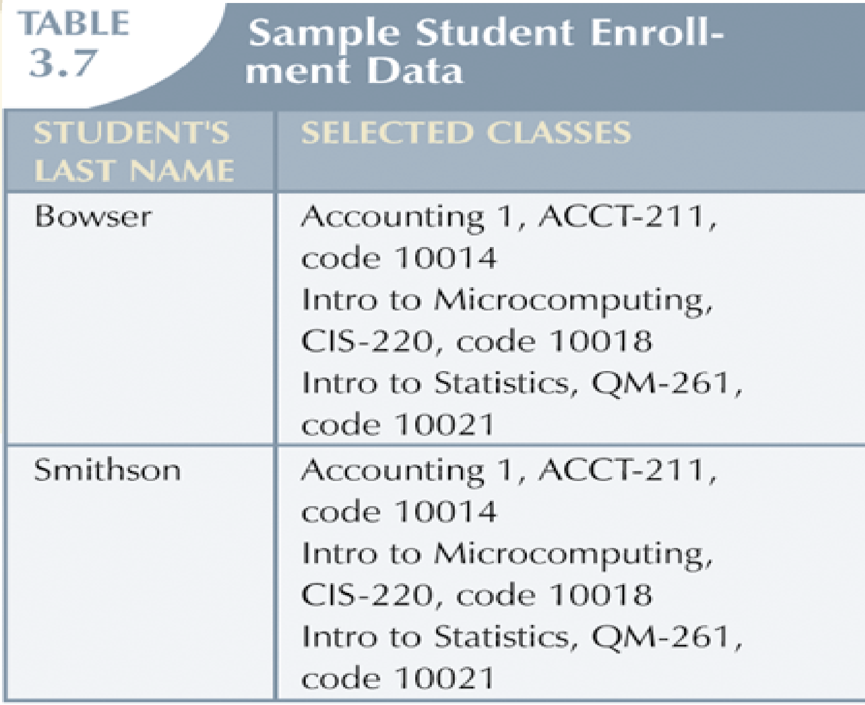


TABLE 3.7 Sample Student Enrollment Data

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

Can the M:N Relationship implemented in relational model ?

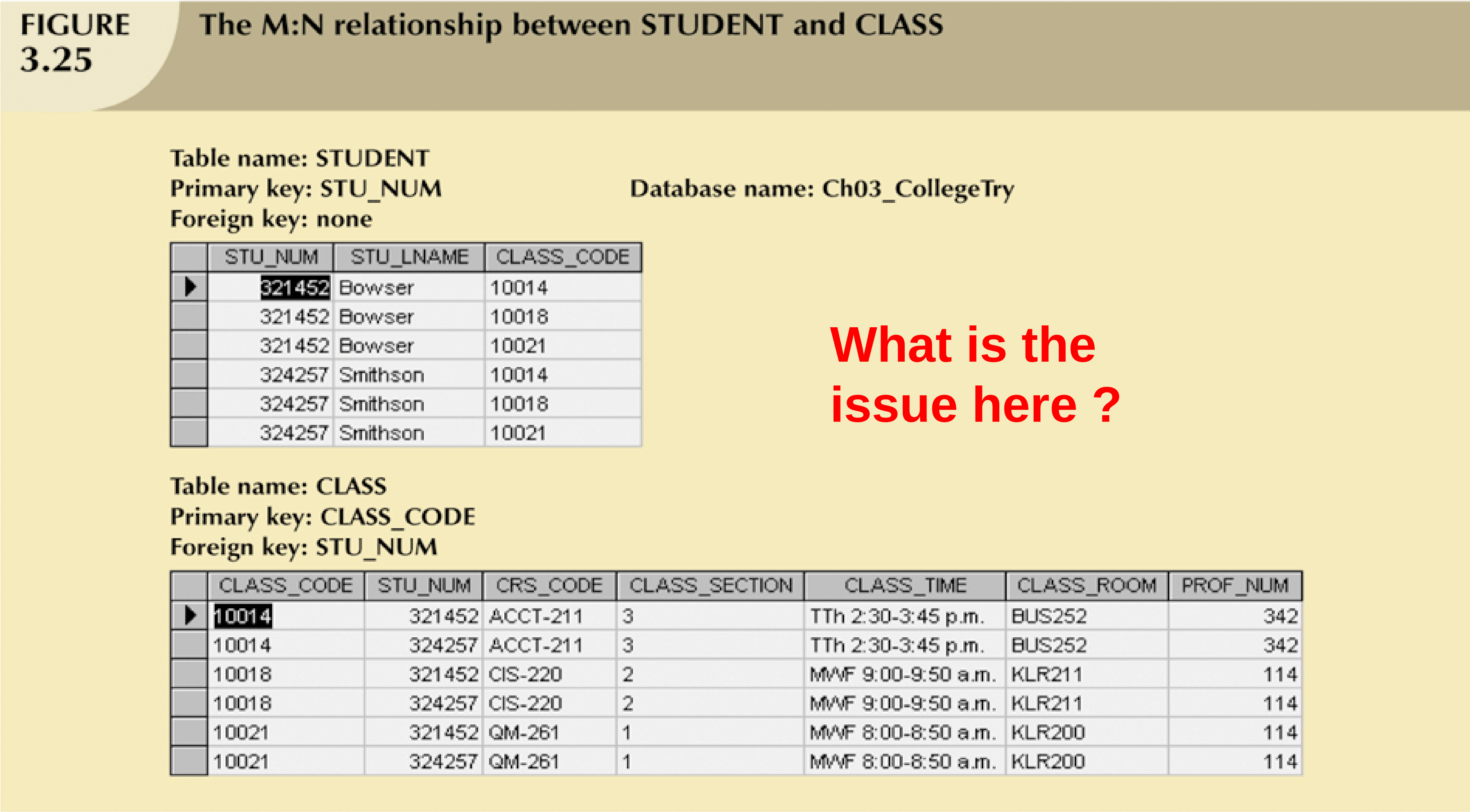


FIGURE 3.25 The M:N relationship between STUDENT and CLASS

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

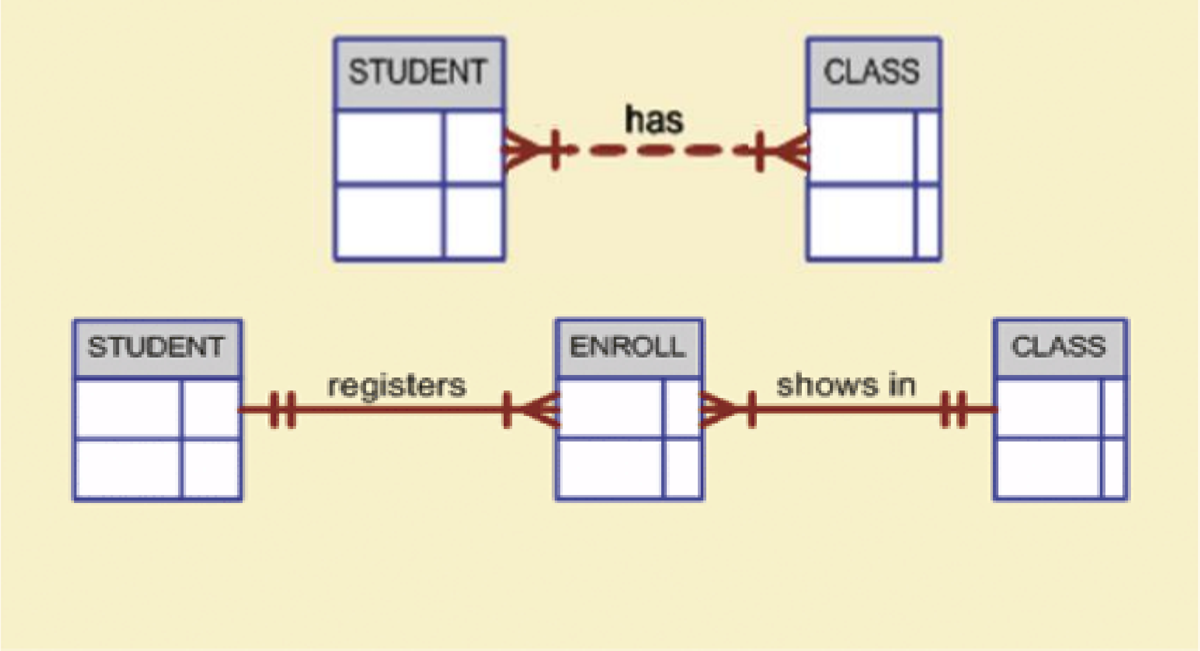
What is the issue here ?

Solution

- **Break M:M to two 1:M relationships**
- Implement the relationship as a composite entity
- 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 3.27 Changing the M:N relationship to two 1:M relationships



The M:N Relationship (continued)

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

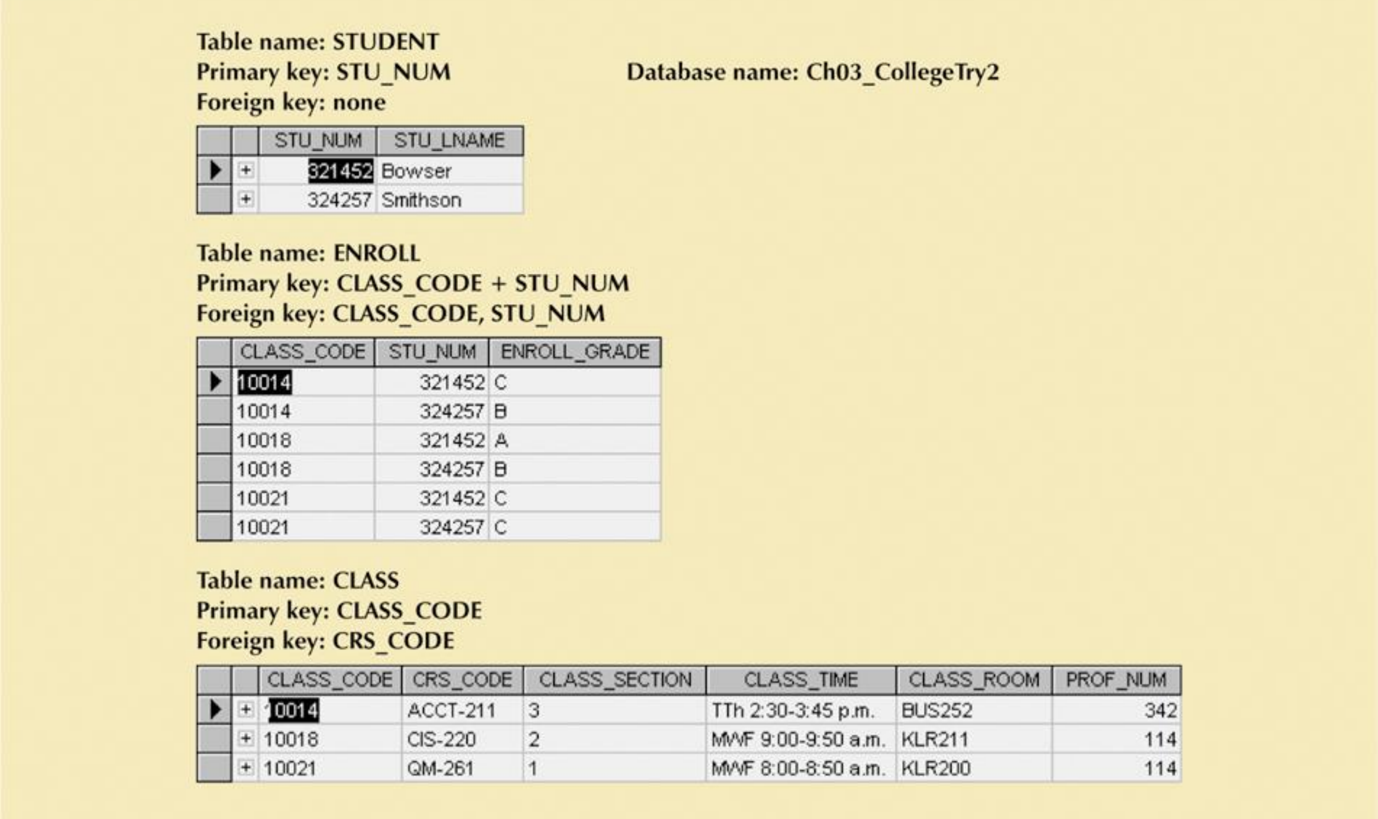


Table name: STUDENT
Primary key: STU_NUM
Foreign key: none

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

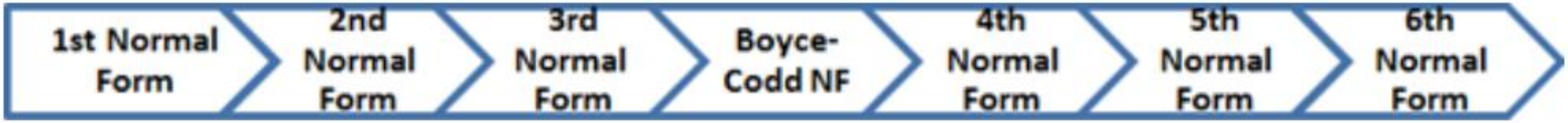
CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

Normalization

- **Normalization** is a database design technique that reduces **data redundancy** and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. (anomaly)
- Normalization rules divides larger tables into smaller tables and links them using relationships.

Normalization (continued)

- The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.
- Works through a series of stages called normal forms:

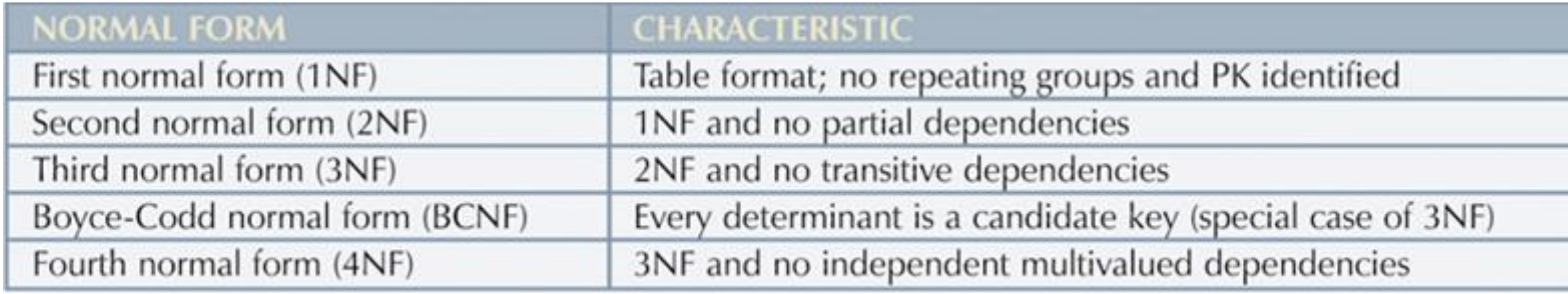


Database Normal Forms

Normalization (continued)

- Normalization (continued)
  - – 2NF is better than 1NF;
  - – 3NF is better than 2NF
  - – For most business database design purposes, **3NF** is as high as we need to go in normalization process
  - – Highest level of normalization is not always most desirable

The Normalization Forms



NORMAL FORM	CHARACTERISTIC
First normal form (1NF)	Table format; no repeating groups and PK identified
Second normal form (2NF)	1NF and no partial dependencies
Third normal form (3NF)	2NF and no transitive dependencies
Boyce-Codd normal form (BCNF)	Every determinant is a candidate key (special case of 3NF)
Fourth normal form (4NF)	3NF and no independent multivalued dependencies

Data Dictionary

- Once we have finalized our relational model (all relations are in 3NF), we need create the **data dictionary** which specifies the details of each of the attributes such as data type, length, whether it is a primary key, foreign key or just an attribute, whether it **must** have value or can be **null** etc.

MS-SQL Data Types – numerical

Data Type	Description	Lower limit	Upper limit	Memory
bigint	It stores whole numbers in the range given	−2^63 (−9,223,372, 036,854,775,808)	2^63−1 (−9,223,372, 036,854,775,807)	8 bytes
int	It stores whole numbers in the range given	−2^31 (−2,147, 483,648)	2^31−1 (−2,147, 483,647)	4 bytes
smallint	It stores whole numbers in the range given	−2^15 (−32,767)	2^15 (−32,768)	2 bytes
tinyint	It stores whole numbers in the range given	0	255	1 byte
bit	It can take 0, 1, or NULL values.	0	1	1 byte/8bit column
decimal	Used for scale and fixed precision numbers	−10^38+1	10^381−1	5 to 17 bytes
money	Used monetary data	−922,337, 203, 685,477.5808	+922,337, 203, 685,477.5807	8 bytes
smallmoney	Used monetary data	−214,478.3648	+214,478.3647	4 bytes

MS-SQL Data Types – date and time

Data Type	**Description**	**Storage size**	**Accuracy**	**Lower Range**	**Upper Range**
DateTime	Used for specifying a date and time from January 1, 1753 to December 31, 9999. It has an accuracy of 3.33 milliseconds.	8 bytes	Rounded to increments of .000, .003, .007	1753-01-01	9999-12-31
smalldatetime	Used for specifying a date and time from January 1, 0001 to December 31, 9999. It has an accuracy of 100 nanoseconds	4 bytes, fixed	1 minute	1900-01-01	2079-06-06
date	Used to store only date from January 1, 0001 to December 31, 9999	3 bytes, fixed	1 day	0001-01-01	9999-12-31
time	Used for storing only time only values with an accuracy of 100 nanoseconds.	5 bytes	100 nanoseconds	00:00:00.0000000	23:59:59.9999999

MS-SQL Data Types – string

Data Type	Description	Lower limit	Upper limit	Memory
char	It is a character string with a fixed width. It stores a maximum of 8,000 characters.	0 chars	8000 chars	n bytes
varchar	This is a character string with variable width	0 chars	8000 chars	n bytes + 2 bytes
varchar (max)	This is a character string with a variable width. It stores a maximum of 1,073,741,824 characters.	0 chars	2^31 chars	n bytes + 2 bytes
text	This is a character string with a variable width. It stores a maximum 2GB of text data.	0 chars	2,147,483,647 chars	n bytes + 4 bytes

Sample Data Dictionary

- Table name: Project

Column Name	Data Type (Length)	Default Value (if any)	Note
Proj_Num	Int	-	Primary Key
Proj_Name	Varchar(100)	-	Must be unique

- Table name: Employee

Column Name	Data Type (Length)	Default Value (if any)	Note
Emp_Num	Integer	-	Primary Key
Emp_Name	Varchar(100)	-	
Job_Class	Varchar(50)		Foreign Key refers to Job table

Sample Data Dictionary

- Table name: Job

Column Name	Data Type (Length)	Default Value (if any)	Note
Job_Class	Varchar(50)		Primary Key
Chg_Hour	Decimal		Hourly rate

- Table name: Assignment

Column Name	Data Type (Length)	Default Value (if any)	Note
Proj_Num	Integer		Primary Key
Emp_Num	Integer	-	Primary Key
Assign_Hours	Decimal	-	Number of hours assigned for the project

Quick Review Question

- What is the difference between relation schema and instance ?
  - Schema – table structure
  - Instance – data in the table
- What is the purpose for primary key ?
  - Uniquely distinguish one row (record) from another row
- What is the function of foreign key ?
  - Establish link (relationship) between two tables
- Can many to many relationship implemented directly in relational model ?
  - Definitely No. Break them into 1:M relationships.

Summary of Main Teaching Points

- Relational model – schema and instance(table)
- Keys – primary and foreign
- Integrity – entity and referential
- Relationship – 1:1, 1:M and M:M
  - M:M → 1:M and M:1
- Normalization

Question and Answer Session

A · P · U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Q & A