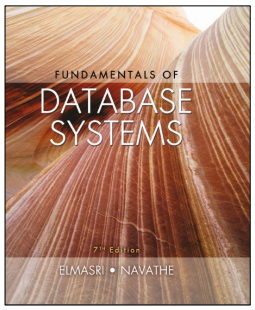
Fundamentals of Database Systems

Seventh Edition



Chapter 3

Data Modeling Using the Entity-Relationship (ER) Model

Pearson

1

Overview of Database Design Process (1 of 2)

- Two main activities:
  - Database design
  - Applications design
- Focus in this chapter on **conceptual database design**
  - To design the conceptual schema for a database application
- Applications design focuses on the programs and interfaces that access the database
  - Generally considered part of software engineering

Pearson

4

Overview of Database Design Process (2 of 2)

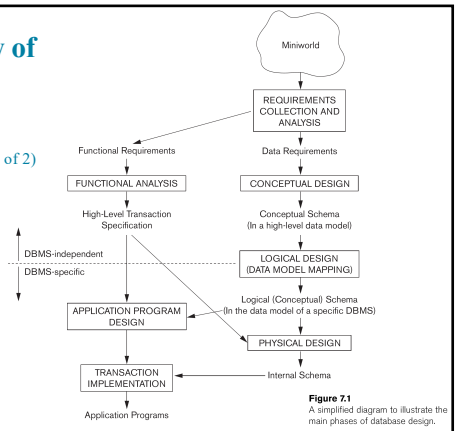


Figure 7.1
A simplified diagram to illustrate the main phases of database design.

Figure 3.1 A simplified diagram to illustrate the main phases of database design.

Pearson

5

Methodologies for Conceptual Design

- Entity Relationship (ER) Diagrams
- Enhanced Entity Relationship (EER) Diagrams
- The UML (Unified Modeling Language) Class Diagrams are popular in industry to document conceptual database designs

Pearson

7

An Example

- UNIVERSITY database
  - Information concerning students, courses, and grades in a university environment
- Data records
  - STUDENT
  - COURSE
  - SECTION
  - GRADE_REPORT
  - PREREQUISITE

Pearson

9

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Pearson

10

Schemas

Figure 2.1
Schema diagram for the database in Figure 1.2.

STUDENT

Name	Student_number	Class	Major

COURSE

Course_name	Course_number	Credit_hours	Department

PREREQUISITE

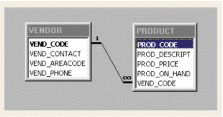
Course_number	Prerequisite_number

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

GRADE_REPORT

Student_number	Section_identifier	Grade



11

An Example (cont'd.)

- Specify structure of records of each file by specifying data type for each data element
  - String of alphabetic characters
  - Integer
  - Etc.

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

Pearson

12

An Example (cont'd.)

- Construct UNIVERSITY database
  - Store data to represent each student, course, section, grade report, and prerequisite as a record in appropriate file
- Relationships among the records
- Manipulation involves querying and updating

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

Pearson

13

An Example (cont'd.)

- Examples of queries:
  - Retrieve the transcript
  - List the names of students who took the section of the 'Database' course offered in fall 2020 and their grades in that section
  - List the prerequisites of the 'Database' course

Pearson

14

An Example (cont'd.)

- Examples of updates:
  - Change the class of 'Smith' to sophomore
  - Create a new section for the 'Database' course for this semester
  - Enter a grade of 'A' for 'Smith' in the 'Database' section of last semester

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Pearson

15

An Example (cont'd.)

- Phases for designing a database:
  - Requirements specification and analysis
  - Conceptual design
  - Logical design
  - Physical design

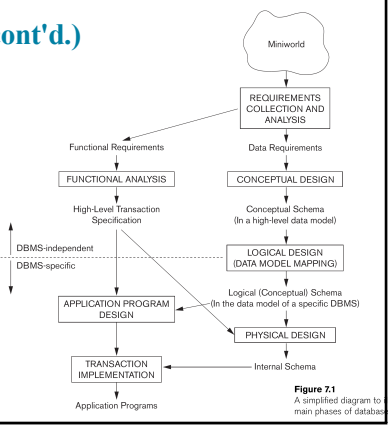


Figure 7.1
A simplified diagram to main phases of databas

Pearson

16

Conceptual Data Modeling Using Entities and Relationships

Logical View of Data

- Relational Database
  - Model enables us to view data *logically* rather than *physically*.
  - Use of table advantageous
    - Structural and data independence
    - Related records stored in independent tables
    - Logical simplicity

18

Entity Types, Entity Sets, Attributes & Keys

- Entity is a *thing* in the real world with an independent existence
- Entity is a person, place, event, or thing about which data is collected

	ENTITIES
Airline environment	pilots
	aircrafts
	routes
	suppliers

- Entities are grouped according to their common characteristics
- The name of the entity reflects its contents

19

Entity Types, Entity Sets, Attributes & Keys (Cont'd.)

- Attributes are characteristics of the entity

	ATTRIBUTES
AIRCRAFT entity	aircraft number
	date of last maintenance
	total hours flown
	hours flown since last maintenance

- Each attribute should be named to remind the user of its content

Entity Types=Relation (Tables)

- A table holds a group of related entity occurrences (entity instances/records)
- consists of rows and columns

20

Entity Types, Entity Sets, Keys, and Value Sets (Domains)

- Entity type
  - Collection (or set) of entities that have the same attributes

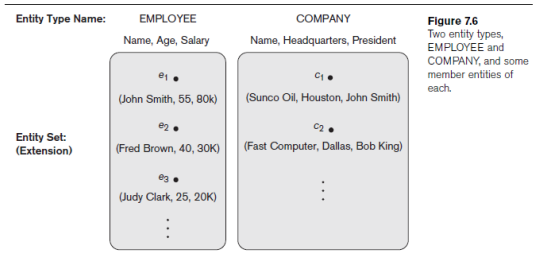


Figure 7.6 Two entity types, EMPLOYEE and COMPANY, and some member entities of each.

21

Relationship Types, Sets, and Instances

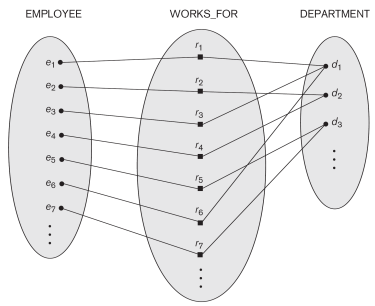


Figure 7.9 Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

22

Characteristics of Relation

- Two-dimensional structure with rows and columns
- Rows (tuples) represent single entity occurrence
- Columns represent attributes and have distinct names
- Row/column intersection represents a single data value
- Tables must have attribute(s) *(primary key)* to uniquely identify each row
- Column values all have same data format
- Each column has range of values called *attribute domain*
- Order of the rows and columns is immaterial to the user

23

Example: Relation STUDENT

Table name: STUDENT

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS
321452	Bowser	William	C	Saturday, February 12, 1972	42	So
324257	Smithson	Anne	K	Tuesday, November 15, 1977	81	Jr
324258	Brewer	Juliette		Tuesday, August 23, 1966	36	So
324269	Oblonski	Walter	H	Sunday, September 16, 1973	66	Jr
324273	Smith	John	D	Friday, December 30, 1955	102	Sr
324274	Katinga	Raphael	P	Thursday, October 21, 1976	114	Sr
324291	Robertson	Gerald	T	Wednesday, April 08, 1970	120	Sr
324299	Smith	John	B	Wednesday, November 30, 1983	15	Fr

STUDENT table, cont.

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_GPA= Grade point average
STU_DOB= Student date of birth
STU_CLASS= Student classification
STU_PHONE= 4-digit campus phone extension
PROF_NUM= Professor number of the professor who is students's advisor

Pearson

24

Keys

The relational database make use of *controlled redundancy* to maintain integrity while linking related tables.

- **Primary Key**
  - An attribute (or group of attributes) selected to uniquely identify all other attribute values in given row.
  - Cannot contain null entries.
- **Determination**
  - "A determines B" indicates that knowing the value of attribute A means that you can look up (determine) the value of attribute B. (A → B)
- **Functional Dependence**
  - If each value in column A determines *one and only one* value in column B, the attribute B is functionally dependent on A.

Pearson

25

Keys (Cont'd.)

- STU_NUM determines STU_LNAME
  STU_NUM → STU_LNAME
  STU_NUM → STU_LNAME, STU_FNAME, STU_DOB
- STU_LNAME does not determine STU_NUM.
  (It is quite possible for several students to be named "Smith")
- If STU_NUM determines only one STU_LNAME;
  STU_LNAME is functionally dependent on STU_NUM.
- STU_NUM determines STU_PHONE.
  (STU_NUM value 321452 determines the STU_PHONE value 2134).
  STU_PHONE is functionally dependent on STU_NUM.
- STU_NUM is not functionally dependent on STU_PHONE.
  (STU_PHONE value 2267 is associated with two STU_NUM values).
- *STU_NUM value 324273 determines the STU_LNAME values Smith.*
  STU_NUM is not functionally dependent on STU_LNAME.
  (STU_LNAME 'Smith' is associated with two STU_NUM values).

Pearson

26

Keys (Cont'd.)

- **Entity integrity**
  - Primary key must uniquely identify each row.
  - A 'null' value is not permitted in the primary key.
- **Foreign Key**
  - An attribute in one table whose values must either match the primary key in another table or be null.
- **Referential integrity**
  - Foreign key must have matching primary key values or nulls.

Pearson

27

FIGURE 3.4 An example of a simple relational database

Database name: Ch03_SaleCo

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

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	€10.23	23	232
123-21UUY	Houselite chain saw, 16 cm bar	€150.09	4	235
QER-34256	Sledge hammer, 16 kg head	€14.72	6	231
SRE-657UG	Rat-tail file	€2.36	15	232
ZZX/3245Q	Steel tape, 12 m length	€5.36	8	235

link

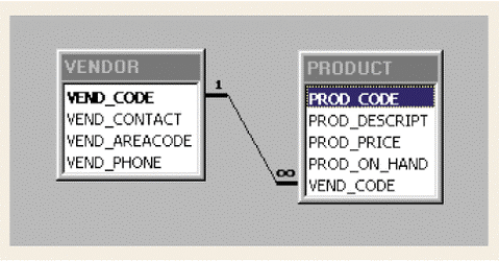
VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	7325	555-1234
231	James Johnson	0181	123-4536
232	Khaya Sibiya	7325	224-2134
233	Lindiwe Molefe	0113	342-6567
234	Nijan Pillay	0181	123-3324
235	Henry Ortozo	0181	899-3425

Table name: VENDOR

Primary key: VEND_CODE

Foreign key: none

28



The Relational Schema

Pearson

29

30

Integrity Rules Revisited

Integrity Type	Requirement	Purpose	Example
ENTITY INTEGRITY	All entries are unique. No null entries in a primary key.	Guarantees that each entity will have a unique identity.	Each car has a unique VIN (Vehicle Identification Number) to insure its proper identification.
REFERENTIAL INTEGRITY	Foreign key must have either a null entry or a matching entry.	Makes it possible for an attribute NOT to have a *corresponding* attribute, but it will be impossible to have an invalid entry.	A customer may not (yet) have an assigned sales representative (number), but it will be impossible to have an invalid sales representative (number).

Pearson

30

FIGURE 3.6 An illustration of integrity rules

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

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	0181	844-2573	12-Mar-19	502
10011	Dunne	Leona	K	0161	894-1238	23-May-18	501
10012	Du Toit	Marlene	W	0181	894-2285	05-Jan-19	502
10013	Pieterse	Jaco	F	0181	894-2180	20-Sep-19	
10014	Orlando	Myron		0181	222-1672	04-Dec-18	501
10015	O'Brian	Amy	B	0161	442-3381	29-Aug-19	503
10016	Brown	James	G	0181	297-1228	01-Mar-19	502
10017	Williams	George		0181	290-2556	23-Jun-19	503
10018	Padayachee	Vinaya	G	1061	382-7185	09-Nov-19	501
10019	Moloi	Mlilo	K	0181	297-3809	18-Feb-19	503

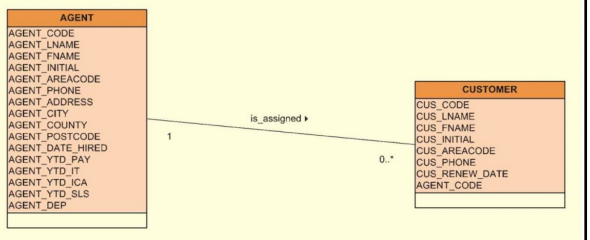
Table name: AGENT
Primary key: AGENT_CODE
Foreign key: none

AGENT_CODE	AGENT_LNAME	AGENT_AREACODE	AGENT_PHONE	AGENT_YTD_SLS
501	Bhengani	0161	228-1249	€1371008.46
502	Mbaso	0181	882-1244	€3923932.59
503	Okon	0181	123-5589	€2444244.52

31

32

FIGURE 2.2 Relational diagram: a relational class diagram



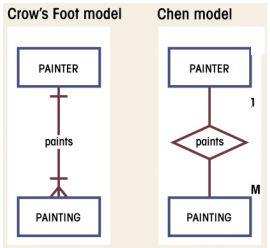
Pearson

32

Conceptual Data Modeling using Entities & Relationships (Cont'd.)

- E-R Model
  - Provides a simplified picture of the relationships among entities
  - ERD is used to map E-R model
  - There are three popular ERD presentation formats;
    - Chen model
    - Crow's Foot model
    - UML ER model

FIGURE 3.7 The 1:* relationship between PAINTER and PAINTING

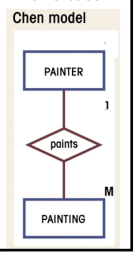


Pearson

39

ERD Display Standards

- Rectangles represent entities
- Entity names are noun, capitalized and singular
- Diamonds represent the relationship(s) between the entities
- Relationship names are active or passive verbs, written in lowercase
- Diamond is connected to the entity rectangles through a relationship line
- "1" side of relationship
  - Number 1 in Chen Model*
  - Bar crossing line in Crow's Foot Model
- "Many" relationships
  - Letter "M" and "N" in Chen Model*
  - Three pronged "Crow's foot" in Crow's Feet Model
- ERD's orientation is immaterial



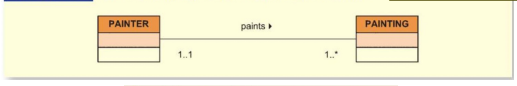
Pearson

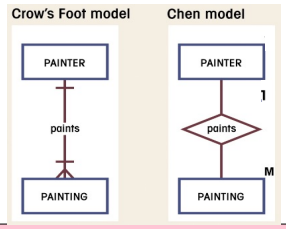
40

Example 1:M Relationship

The UML ER diagram

FIGURE 3.7 The 1:* relationship between PAINTER and PAINTING





A PAINTING is painted by only one PAINTER, but a PAINTER can paint many PAINTINGs.

Pearson

41

Example 1:M Relationship (Cont'd.)

Just make sure that the primary key of the "one table" is included as the foreign key in the "many table".

FIGURE 3.8 The implemented 1:* relationship between PAINTER and PAINTING

Database name: Ch03_Museum Table name: PAINTER
Primary key: PAINTER_NUM Foreign key: none

PAINTER_NUM	PAINTER_LNAME	PAINTER_FNAME	PAINTER_INITIAL
123	Najeke	Onele	P
126	Itero	Julio	G

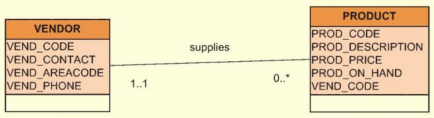
PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126

Table name: PAINTING
Primary Key: PAINTING_NUM Foreign Key: PAINTER_NUM

There may be only one row in the PAINTER table for any given row in the PAINTING table, but there may be many rows in the PAINTING table for any given row in the PAINTER table.

42

FIGURE 3.5 The UML entity relationship diagram for the CH03_SaleCo database



Pearson

43

FIGURE 3.4 An example of a simple relational database

Database name: Ch03_SaleCo

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

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	€10.23	23	232
123-21UUY	Houselite chain saw, 16 cm bar	€150.09	4	235
QER-34256	Sledge hammer, 16 kg head	€14.72	6	231
SRE-657UG	Rat-tail file	€2.36	15	232
ZZX/3245Q	Steel tape, 12 m length	€5.36	8	235

link

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	7325	555-1234
231	James Johnson	0181	123-4536
232	Khaya Sibiya	7325	224-2134
233	Lindiwe Molefe	0113	342-6567
234	Nijan Pillay	0181	123-3324
235	Henry Ortozo	0181	899-3425

Table name: VENDOR

Primary key: VEND_CODE

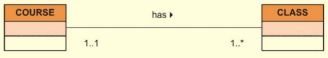
Foreign key: none

44

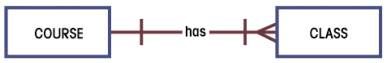
Example 1:M Relationship (Cont'd.)

FIGURE 3.9 The 1:* relationship between COURSE and CLASS

The UML ER diagram



Crow's Foot model



Chen model

Each COURSE can have many CLASSes, but each CLASS references only one COURSE

THE 1:M RELATIONSHIP BETWEEN COURSE AND CLASS

45

Database name: Ch03_TinyUniversity Table name: COURSE
Primary key: CRS_CODE Foreign key: none

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Introduction to Computer Science	3
CIS-420	CIS	Database Design and Implementation	4
QM-261	CIS	Introduction to Statistics	3
QM-362	CIS	Statistical Applications	4

There will be only one row in the COURSE table for any given row in the CLASS table, but there can be many rows in the CLASS table for any given row in the COURSE table.

Table name: CLASS
Primary key: CLASS_CODE Foreign key: CRS_CODE

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	LECT_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

46

Example 1:1 Relationship

FIGURE 3.11 The 1:1 relationship between LECTURER and DEPARTMENT

Pearson

47

Example 1:1 Relationship

The 1:1 LECTURER 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

48

FIGURE 3.12 The implemented 1:1 relationship between LECTURER and DEPARTMENT

Database name: Ch03_TinyUniversity Table name: LECTURER

Primary key: EMP_NUM Foreign key: DEPT_CODE

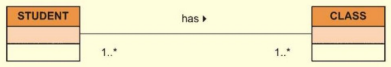
EMP_NUM	DEPT_CODE	LECT_OFFICE	LECT_EXTENSION	LECT_HIGH_DEGREE
103	HIST	DRE 156	6783	PhD
104	ENG	DRE 102	5561	MA
105	ACCT	KLR 229D	8665	PhD
106	MKT/MGT	KLR 126	3899	PhD
110	BIOL	AAK 160	3412	PhD
114	ACCT	KLR 211	4436	PhD
155	MATH	AAK 201	4440	PhD
160	ENG	DRE 102	2248	PhD
162	CIS	KLR 203E	2359	PhD
191	MKT/MGT	KLR 409B	4016	DBA
195	PSYCH	AAK 297	3550	PhD
209	CIS	KLR 333	3421	PhD
228	CIS	KLR 300	3000	PhD
297	MATH	AAK 194	1145	PhD
299	ECON/FIN	KLR 284	2851	PhD
301	ACCT	KLR 244	4683	PhD
335	ENG	DRE 208	2000	PhD
342	SOC	BBG 208	5514	PhD
387	BIOL	AAK 230	8665	PhD
401	HIST	DRE 156	6783	MA
425	ECON/FIN	KLR 284	2851	MBA
435	ART	BBG 185	2278	PhD

The 1:* DEPARTMENT employs LECTURER relationship is implemented through the placement of the DEPT_CODE foreign key in the LECTURER table.

49

Example M:N Relationship

FIGURE 3.13 The *:* relationship between STUDENT and CLASS



Crow's Foot model

Chen model

A STUDENT can take many CLASSes, a CLASS can have many STUDENTs.

Pearson

50

Example M:N Relationship (Cont'd.)

TABLE 3.7 Sample student enrolment data

Student's Last Name	Selected Classes
Ndlovu	Accounting 1, ACCT-211, code 10014 Intro to Computer Science, CIS-220, code 10018 Intro to Statistics, QM-261, code 10021
Smithson	Accounting 1, ACCT-211, code 10014 Intro to Computer Science, CIS-220, code 10018 Intro to Statistics, QM-261, code 10021

Pearson

51

Example M:N Relationship (Cont'd.)

FIGURE 3.14 The *:* relationship between STUDENT and CLASS

It should *not* be implemented as shown in this figure, because the tables create many data redundancies.

Database name: Ch03_CollegeTry Table name: STUDENT

Primary key: STU_NUM Foreign key: none

STU_NUM	STU_LNAME	CLASS_CODE
321452	Ndlovu	10014
321452	Ndlovu	10018
321452	Ndlovu	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

There are many rows in the CLASS table for any given row in the STUDENT table, and there are many rows in the STUDENT table for any given row in the CLASS table. Is it recommended?

52

Converting M:N Relationship to Two 1:M Relationships

FIGURE 3.15 Converting the *:* relationship into two 1:* relationships

Database name: Ch03_CollegeTry2 Table name: STUDENT

Primary key: STU_NUM Foreign key: none

STU_NUM	STU_LNAME
321452	Ndlovu
324257	Smithson

Table name: ENROL

Primary key: CLASS_CODE+STU_NUM

Foreign keys: 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

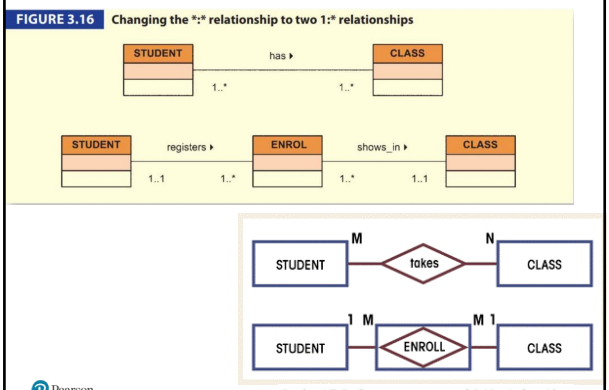
CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

Solution is to create a composite entity (bridge entity). The composite entity includes –as foreign keys- *at least* the primary keys of the tables that are to be linked.

53

Converting M:N Relationship to Two 1:M Relationships

FIGURE 3.16 Changing the *:* relationship to two 1:* relationships



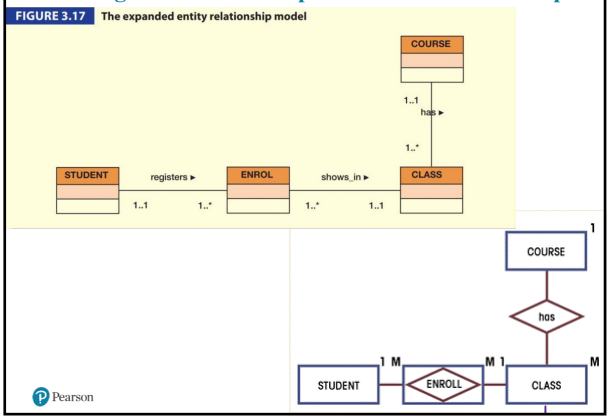
Revised E-R diagram: two sets of 1:M relationships

Pearson

54

Converting M:N Relationship to Two 1:M Relationships

FIGURE 3.17 The expanded entity relationship model

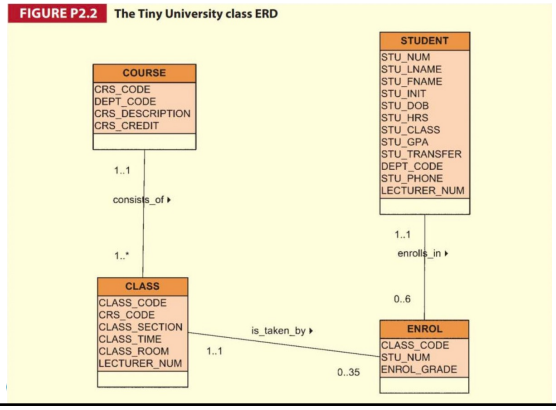


Pearson

55

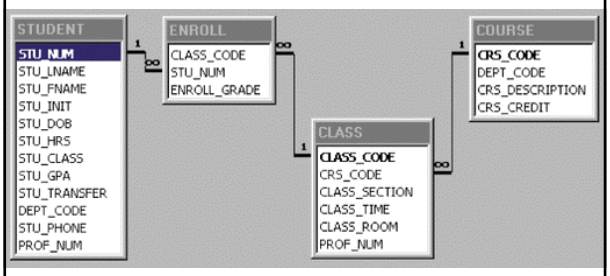
Converting M:N Relationship to Two 1:M Relationships

FIGURE P2.2 The Tiny University class ERD



56

Converting M:N Relationship to Two 1:M Relationships (Cont'd.)



Pearson

Relational Schema

57

Some Points in E-R Modeling

- The orientation of the E-R diagram's components is immaterial
- Avoid crossing relationship lines
- Avoid homonyms - the use of same attribute name to label different attributes
- Avoid synonyms - the use of different names to describe the same attribute
- The dual relationships are a typical indication of the existence of synonyms. Try to remove.

Pearson

58

Some Points in E-R Modeling (Cont'd.)

- Relationships are always read from the 1 to the M side in a 1:M relationship.
- If a 1:1 relationship exists, that relationship is read from the entity that contains the common attributes to the entity that contains the unique attributes. Therefore, EMPLOYEE is a PILOT. (The EMPLOYEE entity contains the common attributes such as names and addresses, while the PILOT entity contains only those attributes – such as medical certificates and pilot licenses that are unique to pilots.)

Pearson

59

Subclasses, Superclasses, and Inheritence

- Enhanced ER (EER) model concepts:

  all the modeling concepts of the ER model *plus*...

  concept of Subclasses, Superclasses (Subtypes, Supertypes)
- EMPLOYEE entity type may be distinguished further into PILOT, TECHNICIAN, SECRETARY, ENGINEER, MANAGER...
- "IS A" relationship → Superclass/Subclass or Supertype/Subtype or Class/Subclass relationship
- An entity that is a member of subclass inherits all the attributes of the entity as a member of the superclass.

Pearson

60

Poor Design vs Good Design

FIGURE 1.3 Employee skill certification in a poor design

ID	EmployeeNumber	EmployeeName	Skill1	Skill1Date	Skill2	Skill2Date	Skill3	Skill3Date
1	02345	Johnny Jones	Basic Database Management	2/14/1993	Advanced Database Management	2/14/1993	Basic Web Design	8/8/1999
2	08273	Marco Blenz	Basic Web Design	3/8/2005	Advanced Process Modeling	8/19/2008		
3	06234	Jasmine Patel	Basic Web Design	8/10/2003	Advanced C# programming	8/10/2003	Basic DB manipulation	1/29/2008
4	03373	Franklin Johnson, Jr.	Advanced Spreadsheets	6/20/2007				
5	13567	Almond, Robert	Basic Process Modeling	9/30/2010	Basic Database Design	5/23/2011		
6	10262	Richardson, Amanda						
7	09382	Jessica Johnson	Basic DB Design	8/2/2008	Basic Database Manipulation	8/2/2008	Advanced DB Manipulation	5/1/2009
8	14311	Duong, Lee	Basic Web Design	9/1/2012				
9			Master Database Programming					
10			Basic Spreadsheets					
11	09002	Ben Joiner	Advanced Spreadsheets	5/16/2009	Basic Web Design	5/16/2009		
12	13383	Raymond F. Matthews	Basic C# Programming	3/12/2010				
13	09283	Chavez, Juan						
14	04893	Patricia Richards	Advanced Database Management	6/11/2002	Advanced Database Manipulation	9/20/2008		
15	13932	Lee, Megan						

SOURCE: Course Technology/Cengage Learning

Pearson

61

Poor Design vs Good Design

FIGURE 1.4 Employee skill certifications in a good design

Database name: Ch01_Text

Table name: EMPLOYEE

Employee_ID	Employee_FName	Employee_LName	Employee_HireDate	Employee_Title
02345	Johnny	Jones	2/14/1993	DBA
03373	Franklin	Johnson	3/15/2000	Purchasing Agent
04893	Patricia	Richards	6/11/2002	DBA
06234	Jasmine	Patel	8/10/2003	Programmer
08273	Marco	Blenz	7/28/2004	Analyst
09002	Ben	Joiner	5/20/2008	Clerk
09283	Juan	Chavez	7/4/2008	Clerk
09382	Jessica	Johnson	8/2/2008	Database Programmer
10262	Amanda	Richardson	4/11/2009	Clerk
13383	Raymond	Matthews	3/12/2010	Programmer
13567	Robert	Almond	9/30/2010	Analyst
13932	Megan	Lee	9/29/2011	Programmer
14311	Lee	Duong	9/1/2012	Programmer

Table name: CERTIFIED

Employee_ID	Skill_ID	Certified_Date
02345	100	2/14/1993
02345	110	8/8/1999
02345	180	2/14/1993
03373	120	6/20/2007
04893	180	6/11/2002
04893	220	9/20/2008
06234	110	8/10/2003
06234	200	8/10/2003
06234	210	1/29/2008
08273	110	3/8/2005
08273	190	8/19/2008
09002	110	5/16/2009
09002	120	5/16/2009
09382	140	8/2/2008
09382	210	8/2/2008
09382	220	5/1/2009
13383	170	3/12/2010
13567	130	9/30/2010
13567	140	5/23/2011
14311	110	9/1/2012

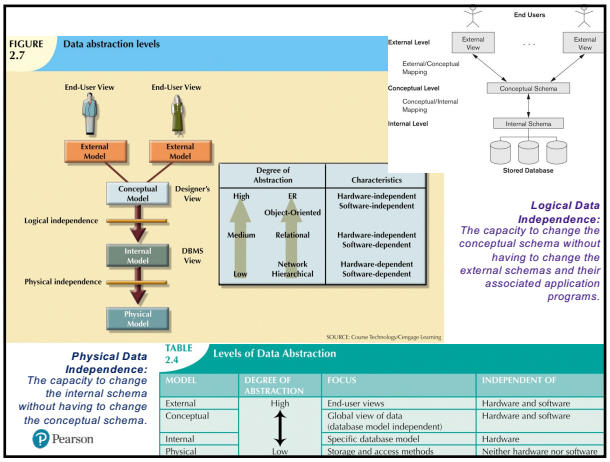
Table name: SKILL

Skill_ID	Skill_Name	Skill_Description
100	Basic Database Management	Create and manage database user accounts.
110	Basic Web Design	Create and maintain HTML and CSS documents.
120	Advanced Spreadsheets	Use of advanced functions, user-defined functions, and macroing.
130	Basic Process Modeling	Create core business process models using standard libraries.
140	Basic Database Design	Create simple data models.
150	Master Database Programming	Create integrated trigger and procedure packages for a distributed environment.
160	Basic Spreadsheets	Create single tab worksheets with basic formulas
170	Basic C# Programming	Create single-tier data aware modules.
180	Advanced Database Management	Manage Database Server Clusters
190	Advanced Process Modeling	Evaluate and Redesign cross-functional internal and external business processes.
200	Advanced C# Programming	Create multi-tier applications using multi-threading
210	Basic Database Manipulation	Create simple data retrieval and manipulation statements in SQL.
220	Advanced Database Manipulation	Use of advanced data manipulation methods for multi-table inserts, set operations, and correlated subqueries.

SOURCE: Course Technology/Cengage Learning

Pearson

62



Logical Data Independence: *The capacity to change the conceptual schema without having to change the external schemas and their associated application programs.*

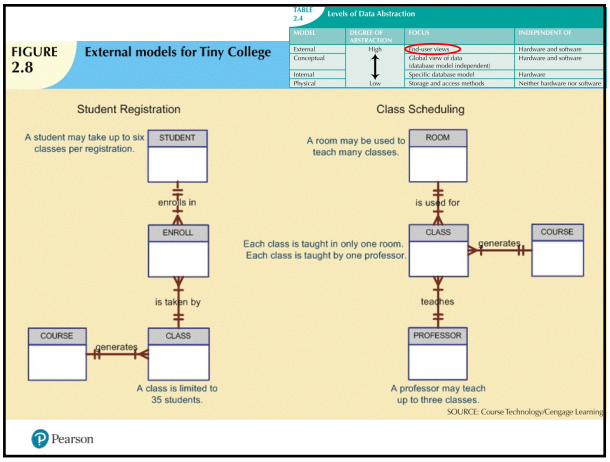
Physical Data Independence: *The capacity to change the internal schema without having to change the conceptual schema.*

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

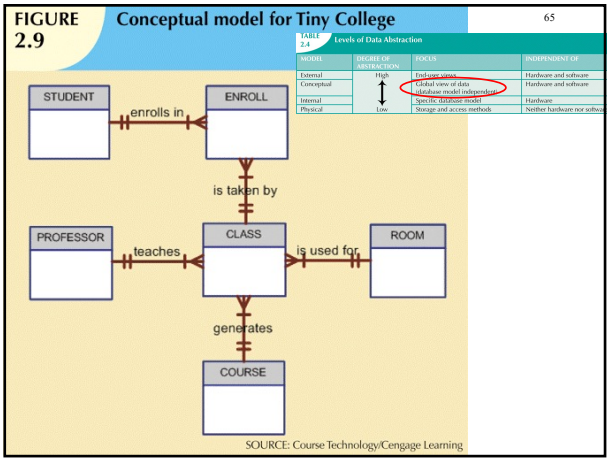
Pearson

63

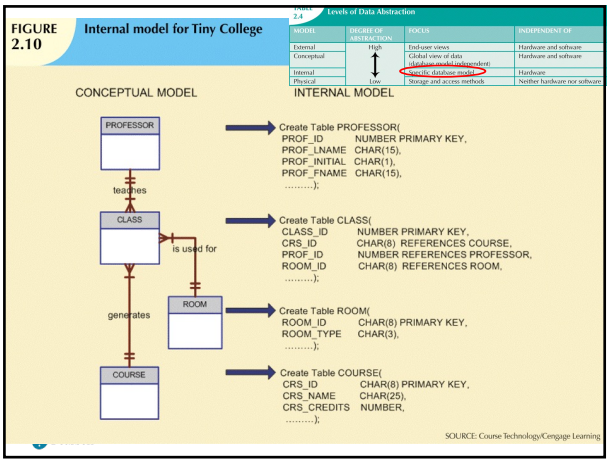


Pearson

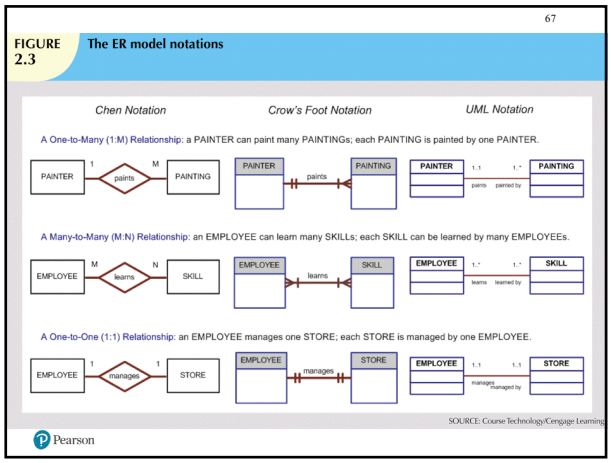
64



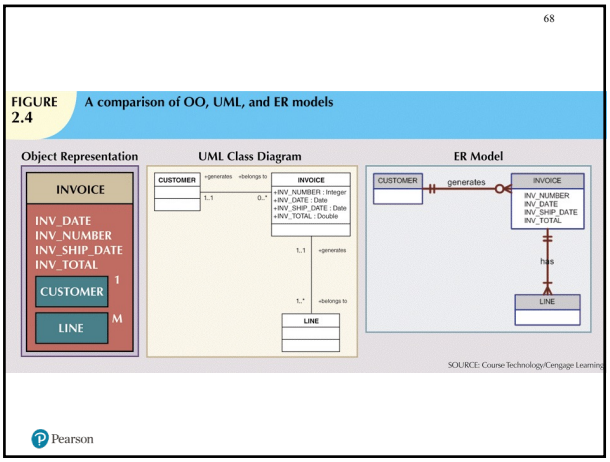
65



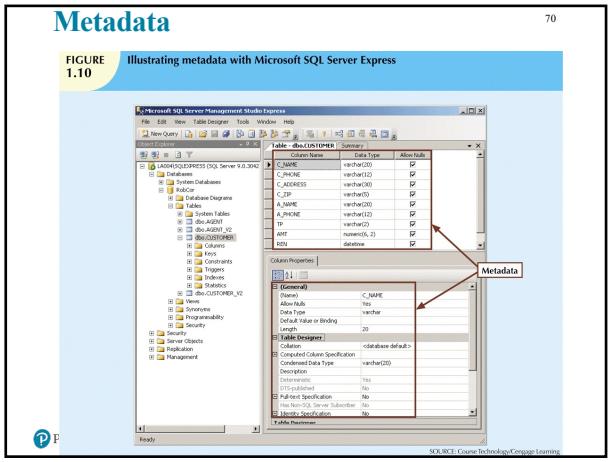
66



67



68



70

Basic Modeling Concepts

- The database design is both art and science
- Good judgement coupled with powerful design tools
- Models
  - "Description or analogy used to visualize something that cannot be directly observed" *Webster's Dictionary*
- Data Model
  - Relatively simple representation of complex real-world data structures
  - Main function is to help us to understand the complexities of real-world environment.
  - Good application --> Good design --> Good model

Pearson

71

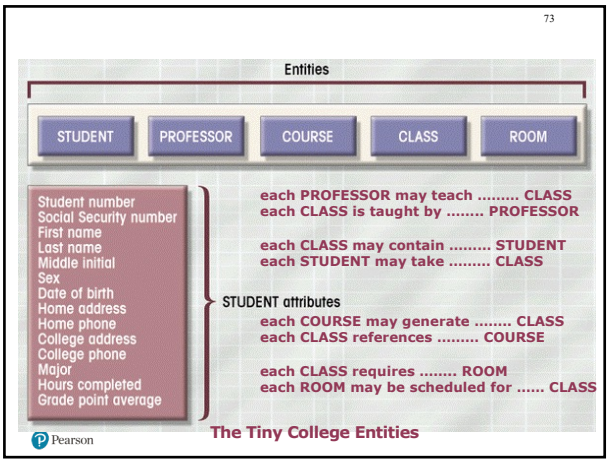
Conceptual Model

- Represents a global view of data
- Basis for identification and description of main data items, avoiding details
- ERD used to represent conceptual data model
- Hardware and software independent

Pearson

72

73

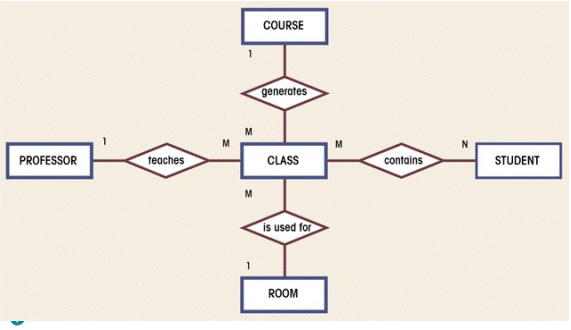


The Tiny College Entities

73

74

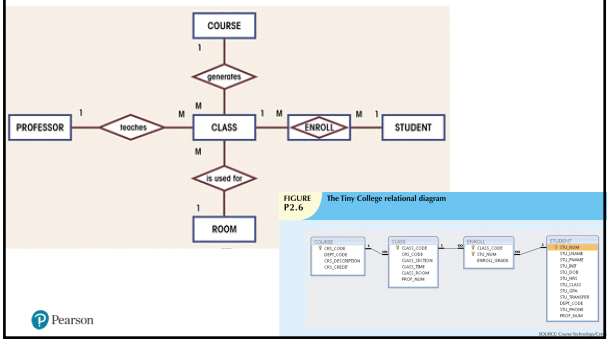
A Conceptual Model for Tiny College



74

75

A Conceptual Model for Tiny College



75

ER Model Concepts

- Entities and Attributes
  - Entity is a basic concept for the ER model. Entities are specific things or objects in the mini-world that are represented in the database.
    - For example the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT
  - Attributes are properties used to describe an entity.
    - For example an EMPLOYEE entity may have the attributes Name, SSN, Address, Sex, BirthDate
  - Each attribute has a **value set** (or data type) associated with it – e.g. integer, string, date, enumerated type, …

Pearson

77

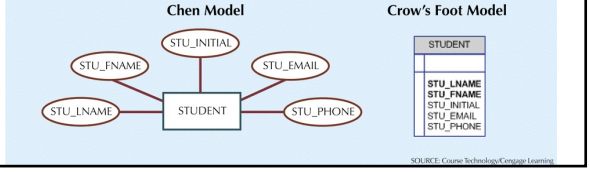
78

Attributes

- Characteristics of entities
- Domain is set of possible values
- Primary keys underlined

TABLE NAME (KEY ATTRIBUTE1, ATTRIBUTE2, ATTRIBUTE 3… ATRIBUTE K)
CAR (CAR_VIN, MOD_CODE, CAR_YEAR, CAR_COLOR)

FIGURE 4.1 The attributes of the STUDENT entity: Chen and Crow's Foot



78

79

Primary Keys

- Ideally, composed of only a single attribute
- Composite key; Primary key composed of more than one attribute

FIGURE 4.2 The CLASS table (entity) components and contents

Database name: Ch04_TinyCollege

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	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
10025	MATH-243	1	Th 6:00-8:40 p.m.	DRE155	325

SOURCE: Course Technology/Cengage Learning

79

Types of Attributes (1 of 2)

- Simple
  - Each entity has a single atomic value for the attribute. For example, SSN or Sex.
- Composite
  - The attribute may be composed of several components. For example:
    - Address(Apt#, House#, Street, City, State, ZipCode, Country), or
    - Name(FirstName, MiddleName, LastName).
    - Composition may form a hierarchy where some components are themselves composite.

Pearson

80

Types of Attributes (2 of 2)

- Multi-valued
  - An entity may have multiple values for that attribute. For example, Color of a CAR or PreviousDegrees of a STUDENT.
    - Denoted as {Color} or {PreviousDegrees}.
- In general, composite and multi-valued attributes may be nested arbitrarily to any number of levels, although this is rare.
  - For example, PreviousDegrees of a STUDENT is a composite multi-valued attribute denoted by {PreviousDegrees (College, Year, Degree, Field)}
  - Multiple PreviousDegrees values can exist
  - Each has four subcomponent attributes:
    - College, Year, Degree, Field

Pearson

81

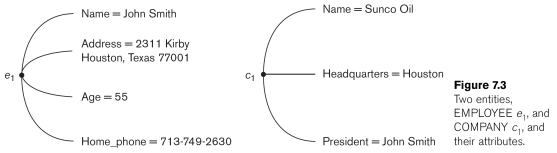


Figure 7.3
Two entities, EMPLOYEE e_1, and COMPANY c_1, and their attributes.

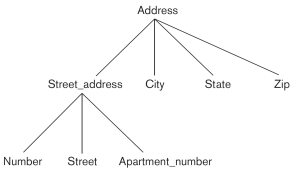


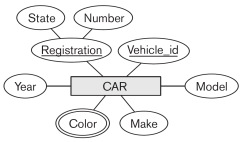
Figure 7.4
A hierarchy of composite attributes.

[9]Zip Code is the name used in the United States for a five-digit postal code, such as 76019, which can be extended to nine digits, such as 76019-0015. We use the five-digit Zip in our examples.

83

Figure 7.7
The CAR entity type with two key attributes, Registration and Vehicle_id. (a) ER diagram notation. (b) Entity set with three entities.

(a)



(b)

CAR
Registration (Number, State), Vehicle_id, Make, Model, Year, {Color}

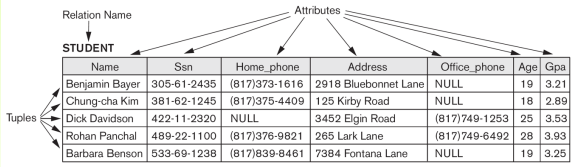
CAR_1
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR_2
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR_3
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

[5]We use a notation for ER diagrams that is close to the original proposed notation (Chen 1976). Many other notations are in use; we illustrate some of them later in this chapter when we present UML class diagrams and in Appendix A.

84



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 3.1
The attributes and tuples of a relation STUDENT.

CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

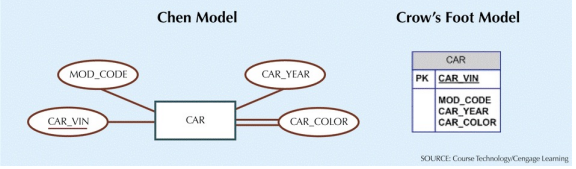
Figure 3.4
The CAR relation, with two candidate keys: License_number and Engine_serial_number.

85

88

Attributes (Con't.)

FIGURE 4.3 A multivalued attribute in an entity



SOURCE: Course Technology/Cengage Learning

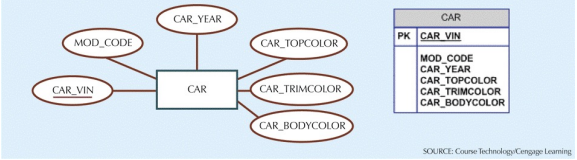
Pearson

88

89

Attributes (Con't.)

FIGURE 4.4 Splitting the multivalued attribute into new attributes

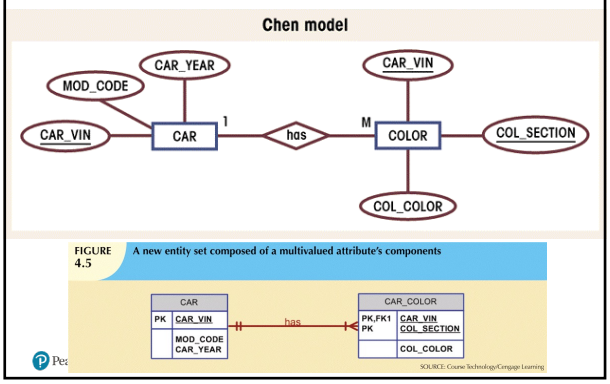


Pearson

89

90

Attributes (Con't.)



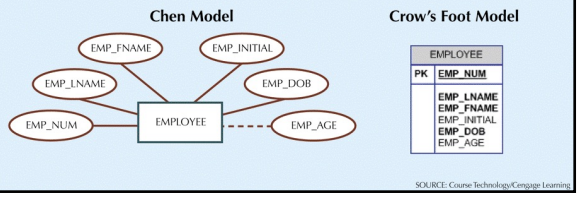
90

91

Attributes (Con't.)

DERIVED ATTRIBUTE
–Need not be physically stored within the database.
–Instead, can be derived by using an algorithm
–Age can be derived from date of birth

FIGURE 4.6 Depiction of a derived attribute



91

92

Relationships

- Association between entities
- Connected entities are called participants
- Operate in both directions
  - Each CUSTOMER may generate many INVOICEs.
  - Each INVOICE is generated by one CUSTOMER.
  - Each DIVISION is managed by one EMPLOYEE
    Ask a question "Can an employee manage more than one division?"
  - Each EMPLOYEE may manage many DIVISIONs (?)
  - Each EMPLOYEE may manage only one DIVISION (?)
- **Connectivity** describes relationship classification
  - 1:1, 1:M, M:N
- **Cardinality Ratios (Multiplicity) for Binary Relationships**
  - Specifies the *maximum* number of relationsip instances that an entity can participate in.

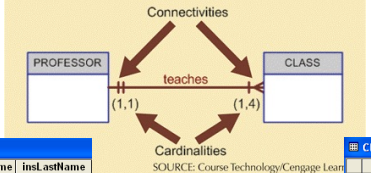
Pearson

92

93

Connectivity and Cardinality Ratios (Multiplicity) in an ERD

FIGURE 4.7 Connectivity and cardinality in an ERD



Professor : Table

insID	insFirstName	insLastName
11	Donna	Bailey
29	Ramone	Baker
31	Terry	Balcombe
33	Greg	McCallum
77	Arnie	Downey
151	Garret	French
159	Rick	Frinksy
171	Patty	Haber

Class : Table

secID	insID
12982	33
12983	33
12998	171
12999	171
13000	159
13001	171
13002	171
13003	33

93

94

Cardinality Ratios (Multiplicity) (Con't.)

- Indicated by placing the appropriate numbers beside the entities, using the format (x, y)
  Minimum value, Maximum value
  - The cardinalities represent the number of occurrences in the *related* entity
  - In the "**PROFESSOR teaches CLASS**" relationship, the cardinality (1,4) written next to the CLASS entity indicates that the PROFESSOR table's foreign key value occurs at least once and no more than four times in the CLASS table
  - If the cardinality had been written as (1,N), there would be no upper limit to the number of classes a professor might teach. The cardinality (1,1) written next to the PROFESSOR entity indicates that each class is taught by one and only one professor. (Each entity occurrence –that is, a row- in the PROFESSOR table can occur once and only once for each professor).

TABLE 4.3 Crow's Foot Symbols

CROW'S FOOT SYMBOLS	CARDINALITY	COMMENT
	(0,N)	Zero or many; the "many" side is optional.
	(1,N)	One or many; the "many" side is mandatory.
	(1,1)	One and only one; the "1" side is mandatory.
	(0,1)	Zero or one; the "1" side is optional.

94

Business Rules

- Statements, derived from a detailed description of the organization's operations
- Define one or more of the following modelling components:
  1. *entities* - in the E-R model corresponds to a table
  2. *relationships* – are associations between entities
  3. *attributes* – are characteristics of entities
  4. *connectivities* – are used to describe the relationship classification
  5. *cardinalities* – express the specific number of entity occurrences associated with one occurrence of the related entity
  6. *constraints* – limitations on the type of data accepted

95

Business Rules (Con't.)

Examples of business rules are:

- An invoice contains one or more invoice lines, but each invoice line is associated with a single invoice.
- A store employs many employees, but each employee is employed by only one store.

Each business rule describes a two-way relationship

- An invoice contains one or more invoice lines.
- Each invoice line is associated with a single invoice.

E-R diagrams cannot always reflect all of the business rules.

- a customer cannot be given a credit line over $10,000 unless that customer has maintained a satisfactory credit history (as determined by the credit manager) during the past two years

is a constraint that cannot be shown in the E-R diagram. The business rule reflected in this constraint would be handled at the applications software level.

96