CHAPTER 6
DATABASE MANAGEMENT SYSTEMS

Management Information Systems, 10th edition,
By Raymond McLeod, Jr. and George P. Schell

Learning Objectives

- Understand the hierarchy of data.
- Learn database structures and how they work.
- Learn how to relate tables together in a database.
- Recognize the difference between a database and a database management system.
- Understand the database concept.
- Learn methods for determining data needs.

Learning Objectives (cont.)

- Understand the basic differences between structured query language and query-by-example.
- Learn about reports and forms.
- Become familiar with entity-relationship diagrams and class diagrams.
- Learn about the important personnel who are associated with databases.
- Learn the advantages and costs of database management systems.

Introduction

- Database management systems organize the large volume of data that firms use in their everyday business activities
- The data organization must also allow managers to find specific data easily and quickly for decision making
- The increased importance of databases as resources supporting decision making has required managers to learn more about database design and use

DATA ORGANIZATION

- Firms need very large amounts of data stored in their computer-based information systems simply because they conduct so many business transactions
- So much data exists that it would be useless for business decision making without an effective and efficient manner for organizing the data
- In order to use the data and avoid chaos, the “data” concept has to be broken down and reduced to smaller concepts
- These smaller concepts of data form the building blocks that can be combined to reproduce the original data in an organized, accessible format

The Data Hierarchy

- Business data is organized into a hierarchy of:
  - data fields that combine to form records; and
  - records that combine to form database files
- A **data field** is the smallest unit of data representing the smallest amount of data that might be retrieved from a computer at a given time
- A **record** is a collection of related data fields
- A **file** is a collection of related records, such as a file of all records containing course codes and title fields
- Table 6.1 shows an example of a database file

Table 6.1

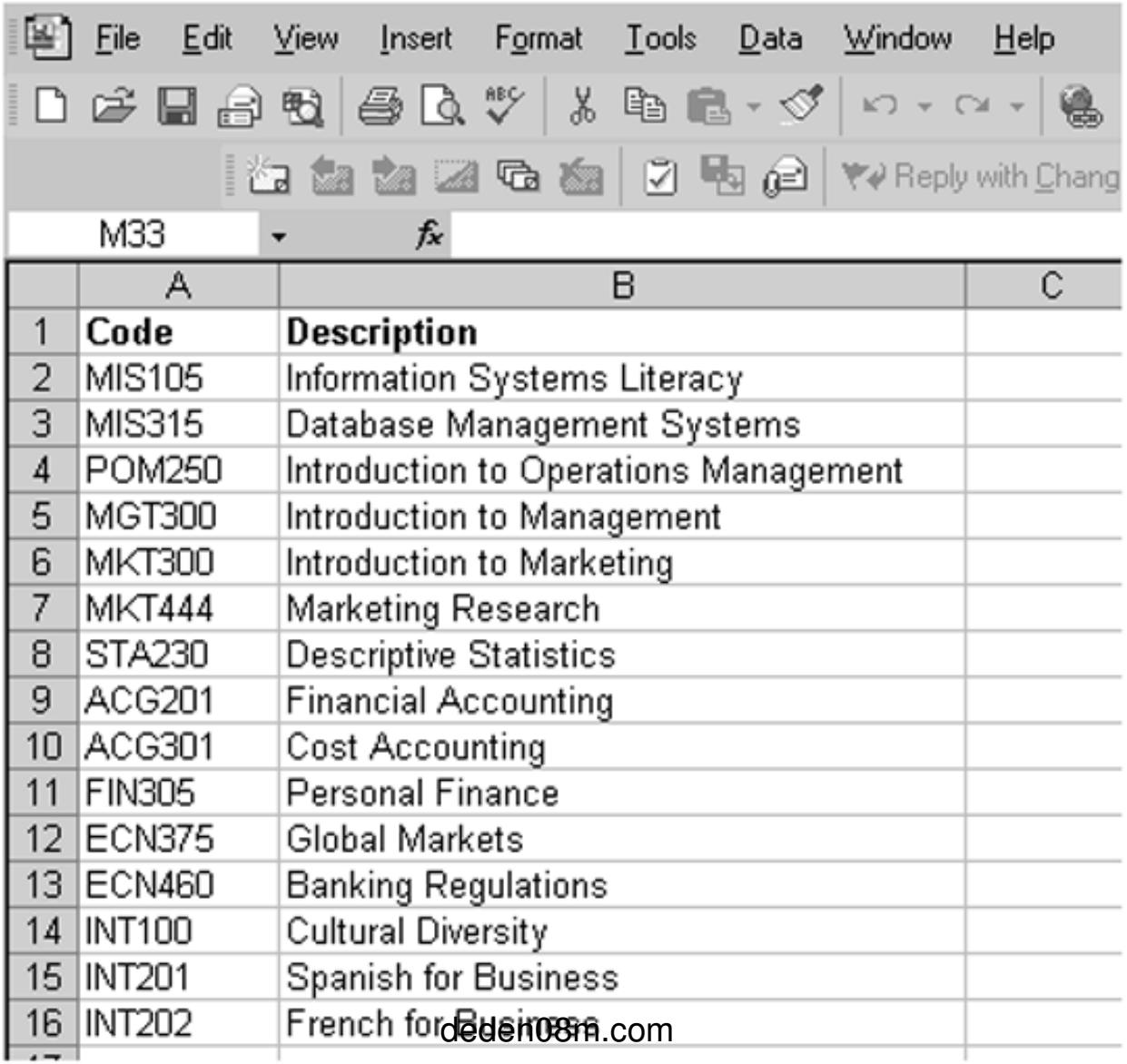
The COURSE Table

CODE	DESCRIPTION
MIS105	Information Systems Literacy
MIS315	Database Management Systems
POM250	Introduction to Operations Management
MGT300	Introduction to Management
MKT300	Introduction to Marketing
MKT444	Marketing Research
STA230	Descriptive Statistics
ACG201	Financial Accounting
ACG301	Cost Accounting
FIN305	Personal Finance
ECN375	Global Markets
ECN460	Banking Regulations
INT100	Cultural Diversity
INT201	Spanish for Business
INT202	French for Business

The Spreadsheet As a Simple Database

- A table of rows and columns can be represented in a spreadsheet
- The columns of the spreadsheet represent the data fields while the column headings contain data field names
- Rows of the table contain the field values
- Figure 6.1 illustrates an Excel spreadsheet containing the values from the COURSE table shown in Table 6.1

Figure 6.1 Spreadsheet Example of the COURSE Table



	A	B	C
1	**Code**	**Description**	
2	MIS105	Information Systems Literacy	
3	MIS315	Database Management Systems	
4	POM250	Introduction to Operations Management	
5	MGT300	Introduction to Management	
6	MKT300	Introduction to Marketing	
7	MKT444	Marketing Research	
8	STA230	Descriptive Statistics	
9	ACG201	Financial Accounting	
10	ACG301	Cost Accounting	
11	FIN305	Personal Finance	
12	ECN375	Global Markets	
13	ECN460	Banking Regulations	
14	INT100	Cultural Diversity	
15	INT201	Spanish for Business	
16	INT202	French for Business	

Flat Files

- A **flat file** is a table that does not have repeating columns
- A flat file provides the constant sequence of data fields that database management requires
- Flat files allow relational database structures to be normalized
- **Normalization** is a formal process for eliminating redundant data fields while preserving the ability of the database to add, modify, and delete records without causing errors

Table 6.2

The COURSE Table with Repeating Columns (Not a Flat File)

AREA	CODE	DESCRIPTION	CODE	DESCRIPTION
MIS	105	Information Systems Literacy	315	Database Management Systems
POM	250	Introduction to Operations Mgt.		
MGT	300	Introduction to Management		
MKT	300	Introduction to Marketing	444	Marketing Research
STA	230	Descriptive Statistics		
ACG	201	Financial Accounting	301	Cost Accounting
FIN	305	Personal Finance		
ECN	375	Global Markets	460	Banking Regulations

Key Fields

- Table 6.3 depicts values in the BOOK table and illustrates the concept of a key
- The **key** in a table is a field (or combination of fields) which contains a value that uniquely identifies each record in the table
- A single field often serves as a key for a table.
- Distinguishing between two or three rows is not enough, key values must be unique for the entire table
- A **candidate key** is a field that uniquely identifies each table row but was not chosen to be the key

Table 6.3

The BOOK Table

ISBN	TITLE
X-15B25	Database Examples
C-12-L	HTML for Beginners
19-63-P	Business Management
ABC-123	Product, Promotion, Placement, and Price
WJY5	Personal Sales Techniques
C-16-MN	Introduction to Accounting
43-U-523	Cost Accounting
HH-7384-GH	Operations Management Fundamentals
322-J	Risk and Returns
32K	Personal Productivity Software
5-53-921	Fundamentals of Hardware
9-7723-K	Stocks Versus Bonds
K-T-127	Human Resources for Today
7-32-881	Oracle Servers
7-32-7723	SQL Servers
A-129-X	Business Management
13-991	The Federal Reserve System
VZ-67	Business French
3-2907-X	Spanish at Work
88-PDQ	Statistics for Business

Key Fields (cont.)

- Some tables require the values of two or more fields to uniquely identify each row in the table
- An example would be when courses have projects
- Table 6.4 shows projects but note that no single data field value uniquely identifies each row
- Values in the *Code* field column repeat between rows. So do field values in all other columns
- The combined values in the *Code* and *Number* fields, however, do form a unique value

Relating Tables

- Sometimes it may be necessary to join tables that originally stand alone
- Consider Table 6.5, the DEPARTMENT table. It shows the six departments offering the courses in the COURSE table
- Note that the tables have no column in common.
- You might be able to guess which department offered each course based upon the values in the *Abbreviation* field but a computer needs an exact match, not a guess
- Table 6.6 depicts the COURSE table with the *Abbreviation* added

Table 6.4

The PROJECT Table

CODE	NUMBER	TITLE	DUE	POINTS
MIS105	1	Home Page Development	9/15/2003	25
MIS105	2	Working With Windows	11/13/2003	50
MIS316	1	Alumni Database	12/5/2003	20
MKT444	1	Finding Customers	10/31/2003	50
MKT444	2	Segmenting Customers	11/21/2003	50
MKT444	3	Customer Service	12/12/2003	40
FIN305	1	Personal Portfolio	11/14/2003	35
INT201	1	Nouns	9/17/2003	15
INT201	2	Verbs	11/21/2003	25
INT202	1	Nouns	9/17/2003	15
INT202	2	Verbs	11/21/2003	25

Table 6.5

The DEPARTMENT Table

ABBREVIATION	NAME	LOCATED	PHONE
ISOM	Information Systems and Operations Management	Cameron Hall	910-3600
MGTMKT	Management and Marketing	Cameron Hall	910-4500
ACGFIN	Accounting and Finance	Dobo Hall	910-1800
ECN	Economics	Randall	910-0900
INT	International Business	Dobo Hall	910-0900

Table 6.6

The COURSE Table with Abbreviation Field Added

CODE	DESCRIPTION	ABBREVIATION
MIS105	Information Systems Literacy	ISOM
MIS315	Database Management Systems	ISOM
POM250	Introduction to Operations Management	ISOM
MGT300	Introduction to Management	MGTMKT
MKT300	Introduction to Marketing	MGTMKT
MKT444	Marketing Research	MGTMKT
STA230	Descriptive Statistics	ISOM
ACG201	Financial Accounting	ACGFIN
ACG301	Cost Accounting	ACGFIN
FIN305	Personal Finance	ACGFIN
ECN375	Global Markets	ECN
ECN460	Banking Regulations	ECN
INT100	Cultural Diversity	INT
INT201	Spanish for Business	INT
INT202	French for Business	INT

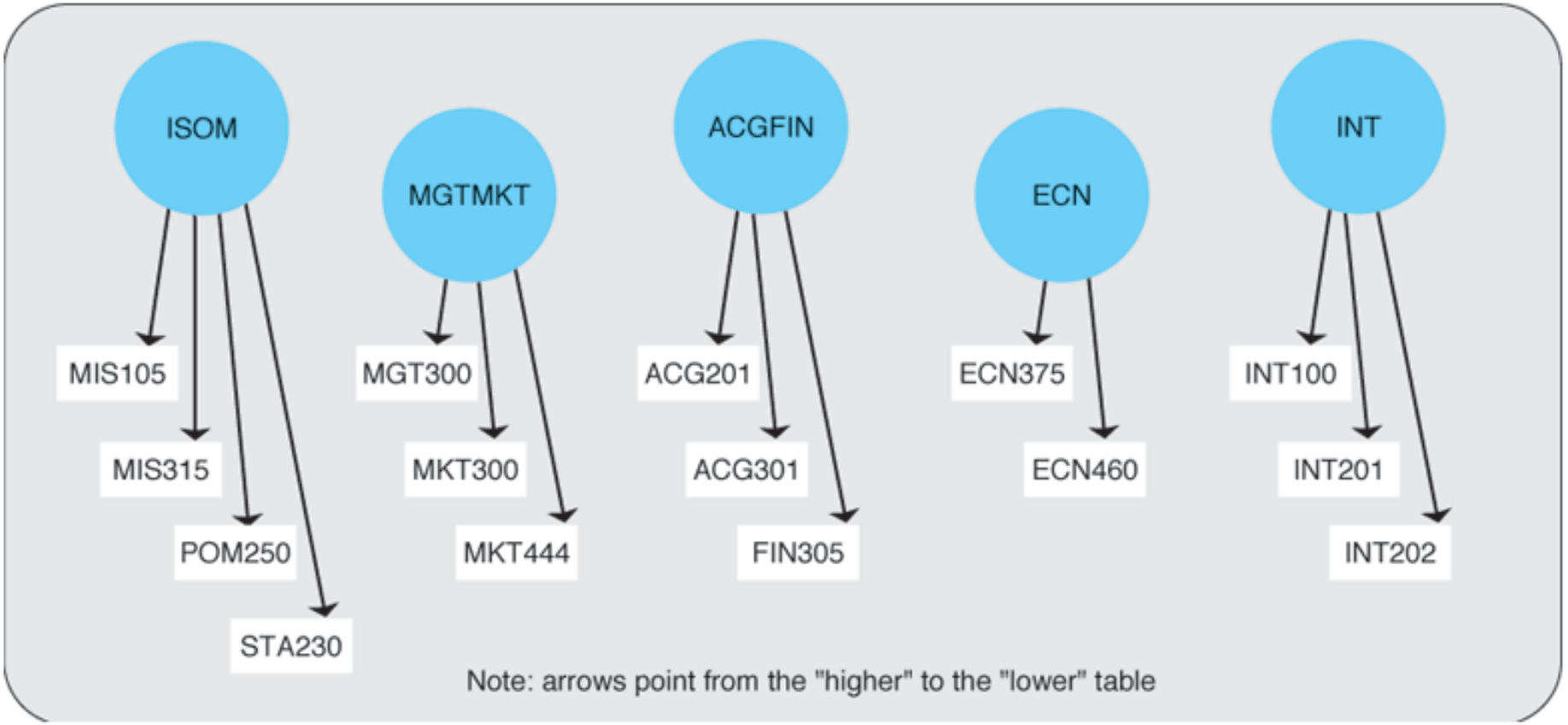
DATABASE STRUCTURES

- Database structures are ways of organizing data in order to make data processing more efficient
- The structure is then implemented via a **database management system (DBMS)** which is a software application that:
  - stores the structure of the database;
  - stores the data itself;
  - stores the relationships among data in the database;and
  - forms and reports pertaining to the database including the data field description
- Because it contains the data field definitions, the database controlled by a DBMS is called a "self-describing set of related data"

Hierarchical Database Structures

- The IDS database management system was one of the first DBMS and conformed to the **hierarchical database structure**
- The hierarchical structure is formed by data groups, subgroups, and further subgroups
- Figure 6.2 shows navigation from the DEPARTMENT table to the COURSE table using a hierarchical database structure

Figure 6.2 The Hierarchical Structure between the DEPARTMENT and COURSE Tables

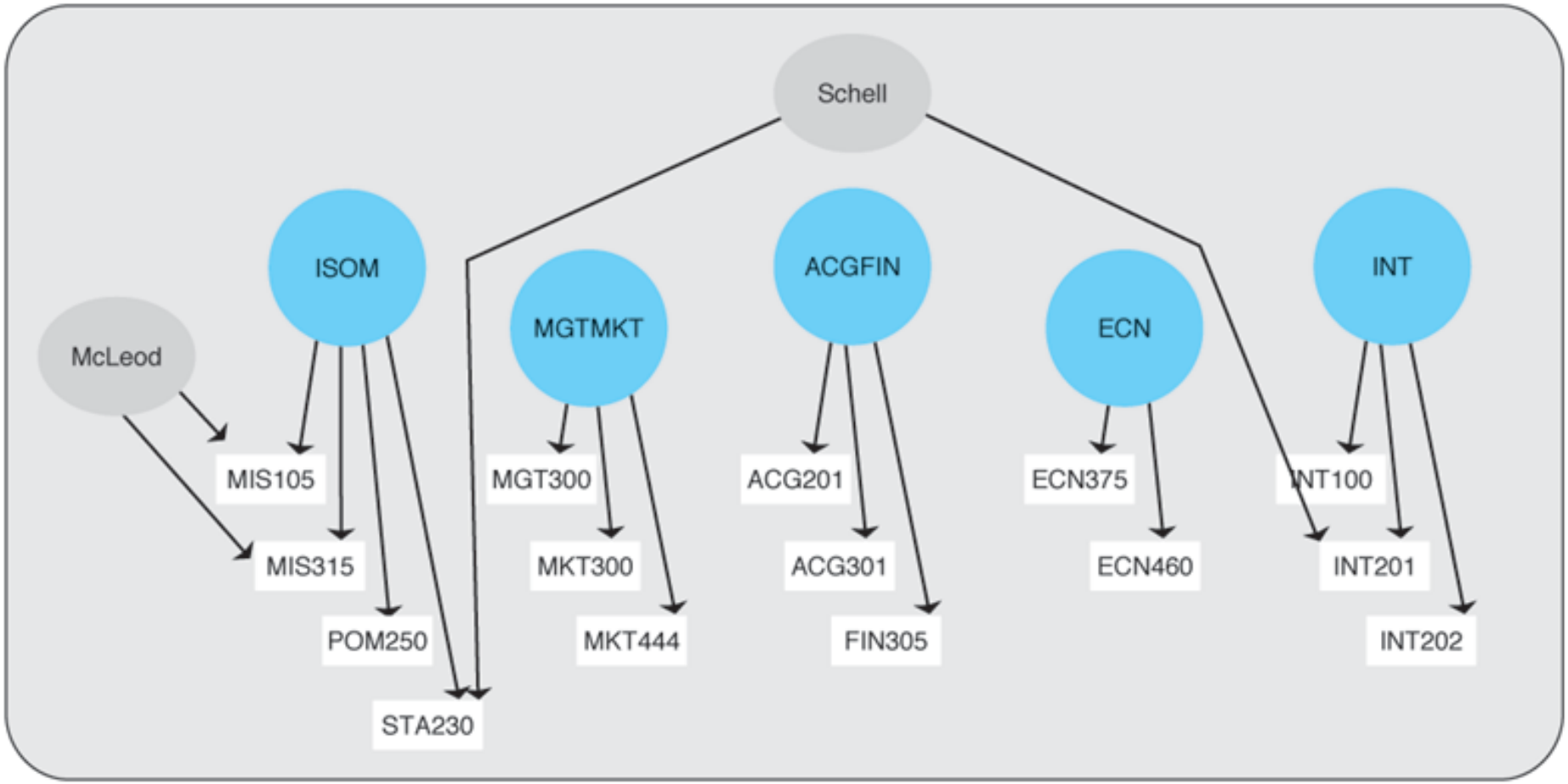


Network Database Structures

- Network database structures were developed to allow retrieval of specific records
- They allow any given record to point to any other record in the database
- Networks solve the problem of having to backtrack all the way to a joining “branch” of the database
- However, this wide range of possible connections is also the weakness of applying network structures to practical problems since it was just too complex to allow every record to point to every other record

- The breakthrough came from basic research conducted independently by C. J. Date and E. F. Codd using relational algebra
- They were able to show that relational databases created out of a series of interrelated tables were, in fact, far more flexible and versatile than either the hierarchical or network database structures
- Whereas the hierarchical and network database structures rely on **physical relationships** in the form of storage addresses, relational database structures use **implicit relationships** that can be implied from the data (see Figure 6.3)

Figure 6.3 Adding a Table for FACULTY Is Beyond the Ability of Hierarchical Database Structures



A RELATIONAL DATABASE EXAMPLE

- A database named *Schedule* has been created from tables used earlier in the chapter and some others
- The database is implemented in Microsoft Access 2002 (also known as Access XP).
- Databases break information into multiple tables because if information were stored in a single table, many data field values would be duplicated

- The example is implemented on Microsoft Access dbms but would be similar on any relational dbms product
- The COURSE table in Access (Figure 6.4) is a list of data field values. The table itself had to be defined in Access before values were entered into the data fields
- Figure 6.5 shows the definition of the *Code* field
- Figure 6.6 illustrates that *Abbreviation* field values will be looked up from a list of values in the DEPARTMENT table
- Table 6.7 shows a single table of course and department fields before they were separated into different tables

Figure 6.4 The COURSE Table in Access

COURSE : Table

Code	Description	Abbreviation
ACG201	Financial Accounting	ACGFIN
ACG301	Cost Accounting	ACGFIN
ECN375	Global Markets	ECN
ECN460	Banking Regulations	ECN
FIN305	Personal Finance	ACGFIN
INT100	Cultural Diversity	INT
INT201	Spanish for Business	INT
INT202	French for Business	INT
MGT300	Introduction to Management	MGTMKT
MIS105	Information Systems Literacy	ISOM
MIS315	Database Management Systems	ISOM
MKT300	Introduction to Marketing	MGTMKT
MKT444	Marketing Research	MGTMKT
POM250	Introduction to Operations Management	ISOM
STA230	Descriptive Statistics	ISOM

Figure 6.5 Defining the Code Field in the COURSE Table

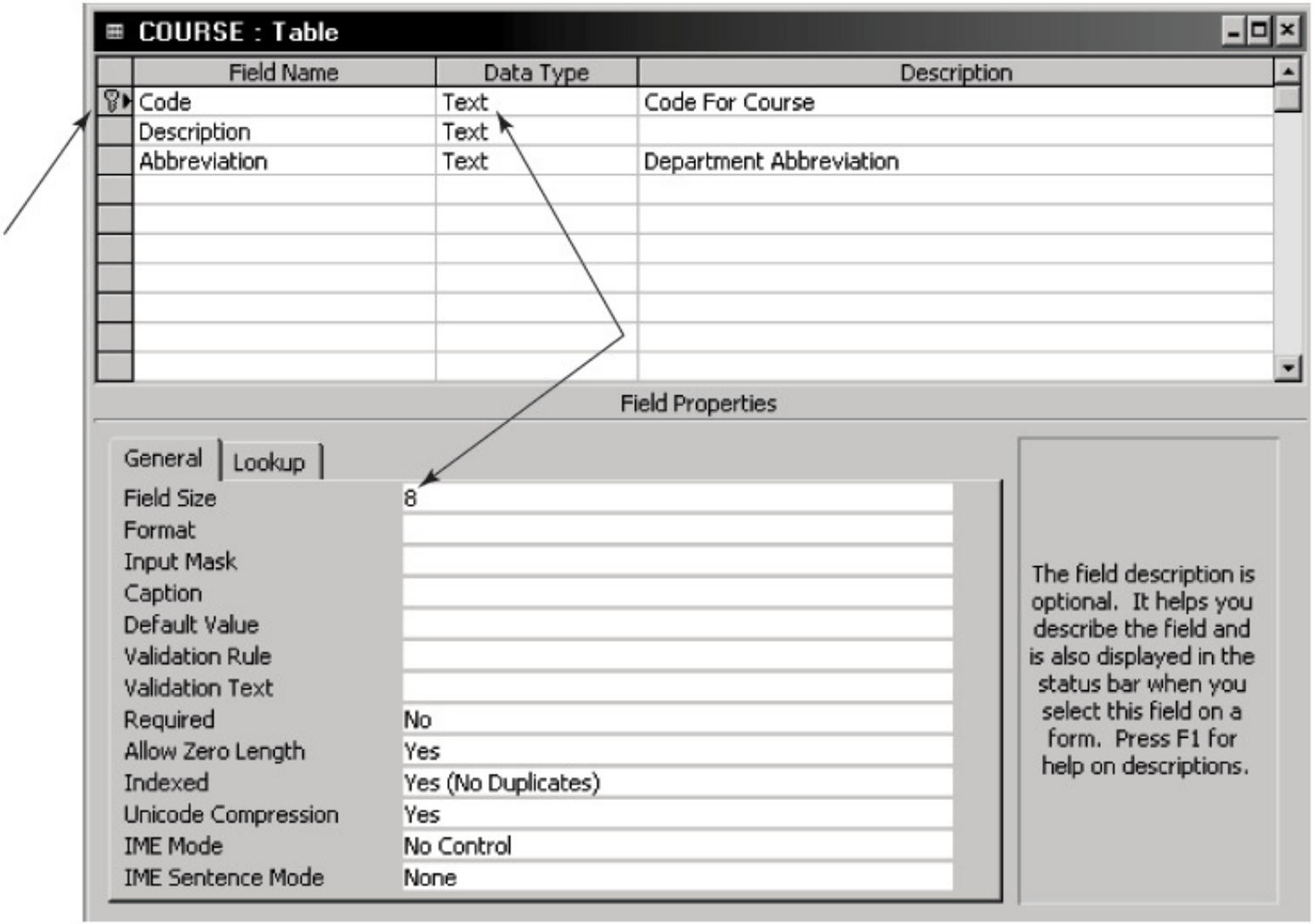


Figure 6.6 Look-Up Values

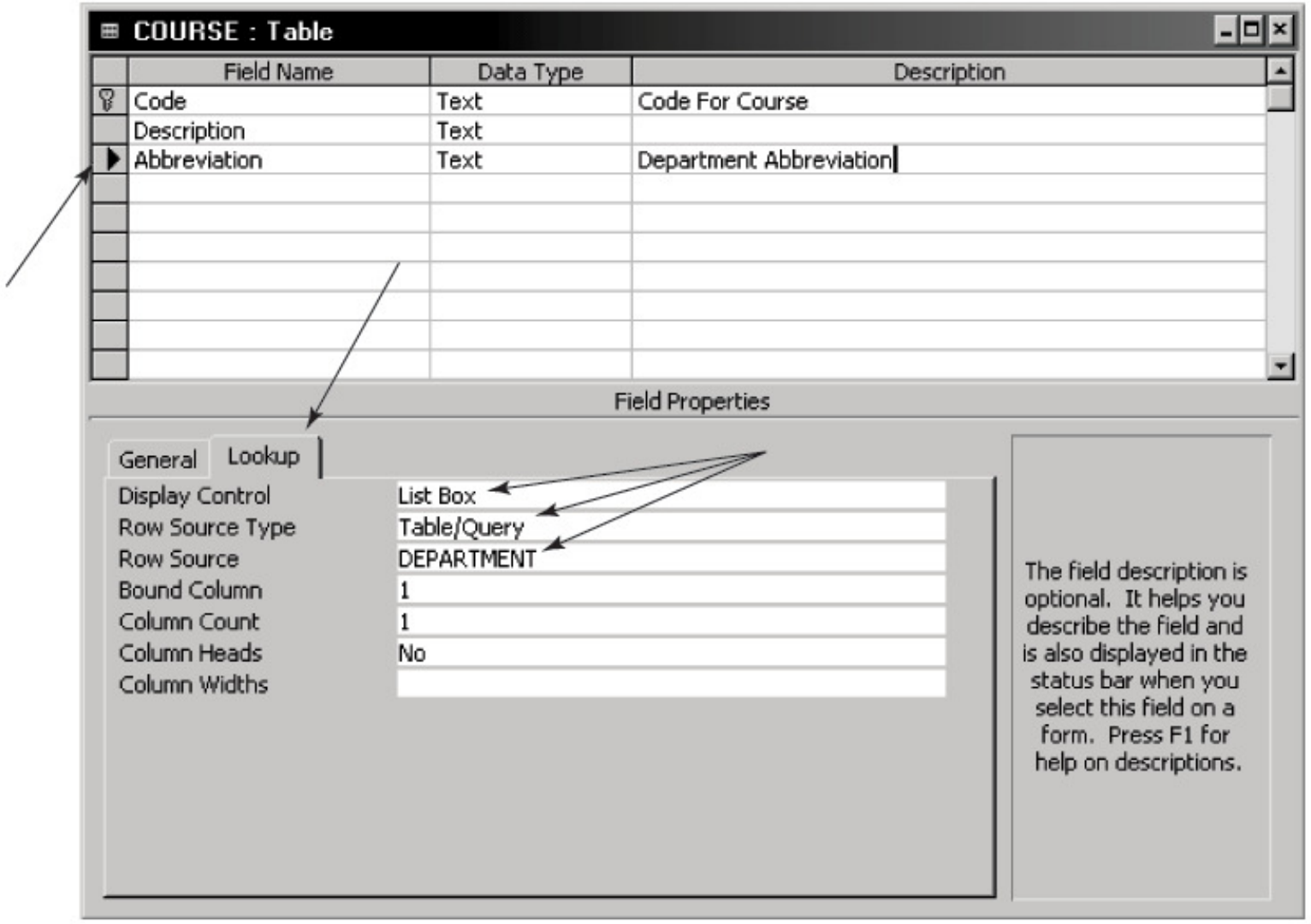
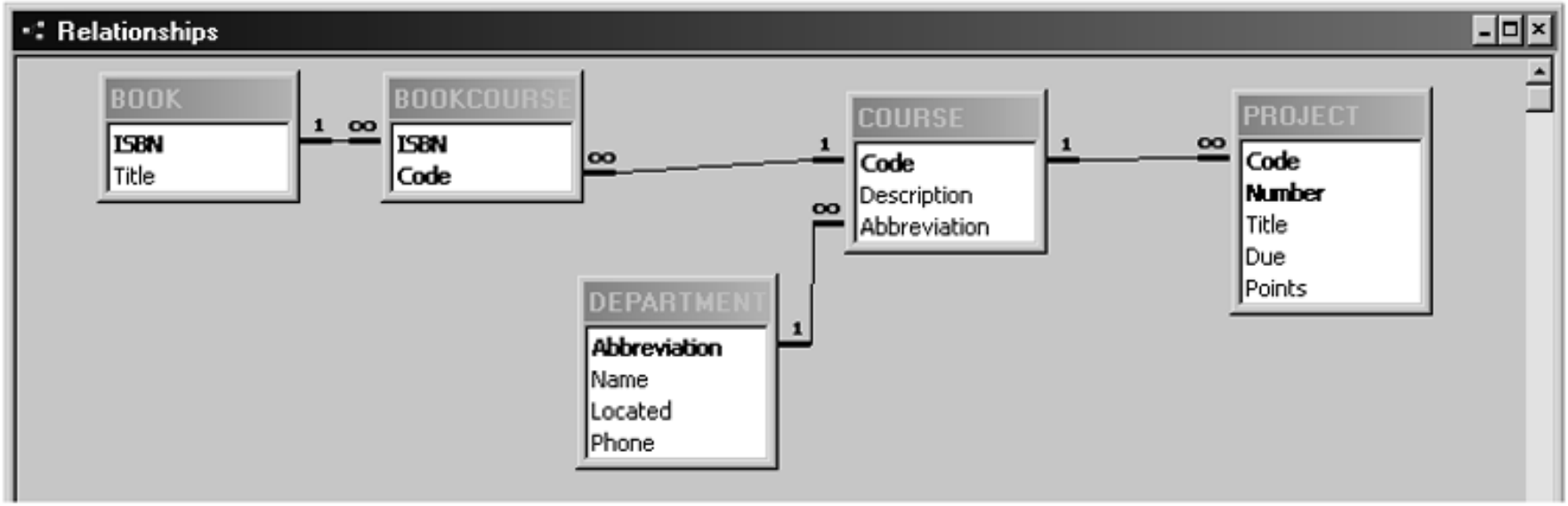


Table 6.7

Unseparated Table of Course and Department Data Fields

Code	Description	Abbreviation	Name	Located	Phone
MIS105	Information Systems Literacy	ISOM	Information Systems and Operations Management	Cameron Hall	910-3600
MIS315	Database Management Systems	ISOM	Information Systems and Operations Management	Cameron Hall	910-3600
POM250	Introduction to Operations Management	ISOM	Information Systems and Operations Management	Cameron Hall	910-3600
MGT300	Introduction to Management	MGTMKT	Management and Marketing	Cameron Hall	910-4500
MKT300	Introduction to Marketing	MGTMKT	Management and Marketing	Cameron Hall	910-4500
MKT444	Marketing Research	MGTMKT	Management and Marketing	Cameron Hall	910-4500
STA230	Descriptive Statistics	ISOM	Information Systems and Operations Management	Cameron Hall	910-3600
ACG201	Financial Accounting	ACGFIN	Accounting and Finance	Dobo Hall	910-1800
ACG301	Cost Accounting	ACGFIN	Accounting and Finance	Dobo Hall	910-1800
FIN305	Personal Finance	ACGFIN	Accounting and Finance	Dobo Hall	910-1800
ECN375	Global Markets	ECN	Economics	Randall	910-0900
ECN460	Banking Regulations	ECN	Economics	Randall	910-0900
INT100	Cultural Diversity	INT	International Business	Dobo Hall	910-0900
INT201	Spanish for Business	INT	International Business	Dobo Hall	910-0900
INT202	French for Business	INT	International Business	Dobo Hall	910-0900

Figure 6.7 Access View of Tables, Fields, and Their Relationships



- The logical integration of records across multiple physical locations is called the **database concept.** It is not dependent on the user's perception of logical location
- Two primary goals of the database concept are to minimize data redundancy and to achieve data independence
- **Data independence** means placing the data specifications in tables and dictionaries that are physically separate from the programs
- **Data dictionary** refers to the definition of data stored within the database and controlled by the DBMS

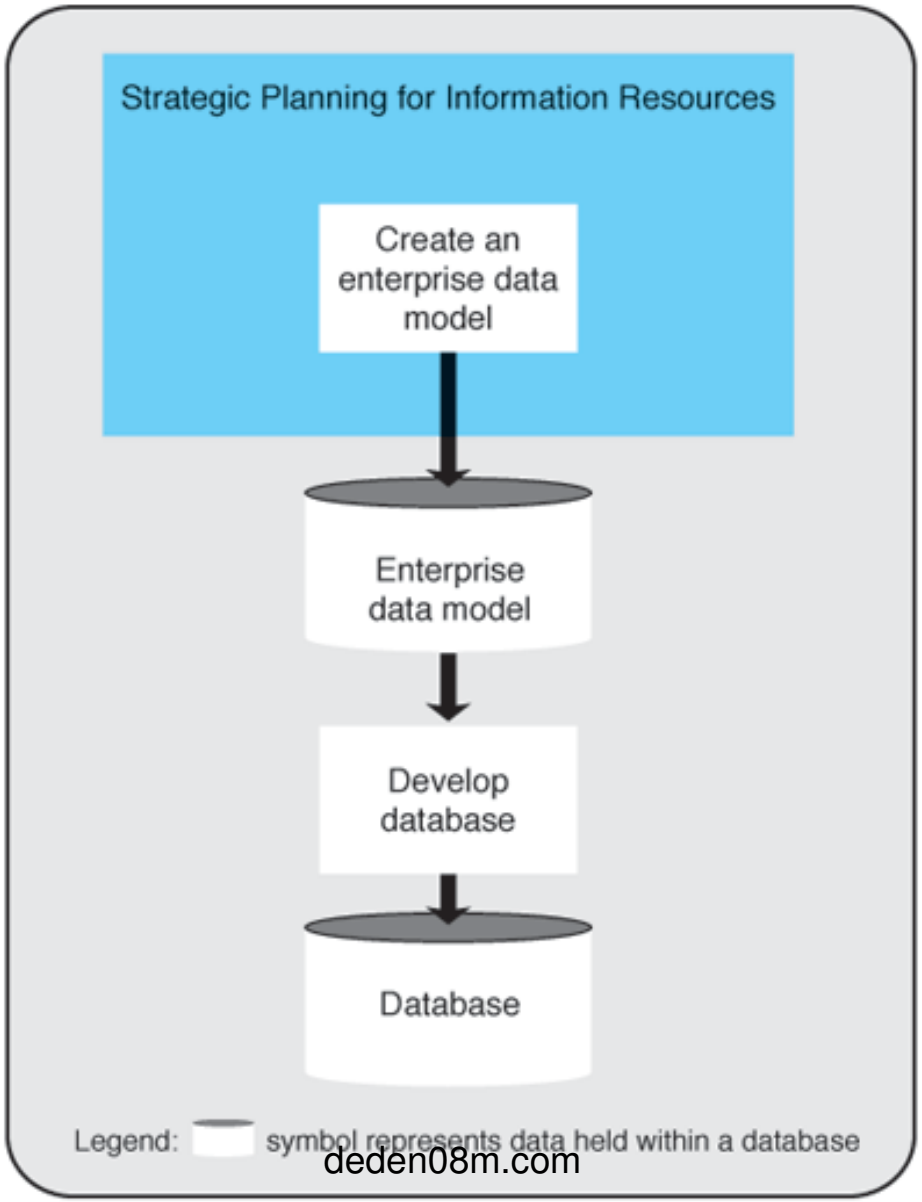
CREATING A DATABASE

- In the conceptual model you:
  - Determine the data that you need
  - Describe the data
  - Enter the data into the database

The two main approaches to determine data needs are:

- The Process-Oriented Approach:
  - 1. The *problem* is defined
  - 2. The *decisions* required to solve the problem are identified
  - 3. For each decision the required *information* is described
  - 4. The *processing* necessary to produce the information is determined
  - 5. The *data* required by the processing is specified
- The Enterprise Modeling Approach
  - the firm's entire data needs are determined and then stored in the database
- The **enterprise data model** is shown in Figure 6.8

Figure 6.8 Creating an Enterprise Data Model



- Modeling the firm's data needs is supported by techniques that:
    - Describe the data
    - Describe how the data aggregates into tables
    - Describe how tables relate to each other
- Entity-relationship diagrams are used to describe relationships between conceptual collections of data so that their related records can be joined together
- Class diagrams are used to describe both the data relationships and the actions that operate on the data in the relationships

Entity-Relationship Diagrams

- **ER Diagrams** deal with data in **entities** (conceptual collections of related data fields) and the relationships between entities
- If we need to describe the data needed for a new information system to keep track of firms and their employees as well as their products, we can imagine that three separate data entities will exist: **firm**, **employee**, and **product** (Figure 6.9)
- When firms hire employees, however, there is an independent relationship between those two entities (Figure 6.10)
- Figure 6.11 demonstrates how we specify that one record in the firm entity can be related to many records in the product entity and also that one record in the firm entity can relate to many records in the employee entity
- Figure 6.12 is a "many-to-many" example

Figure 6.9 Entities

Figure 6.10 Entities and Relationships

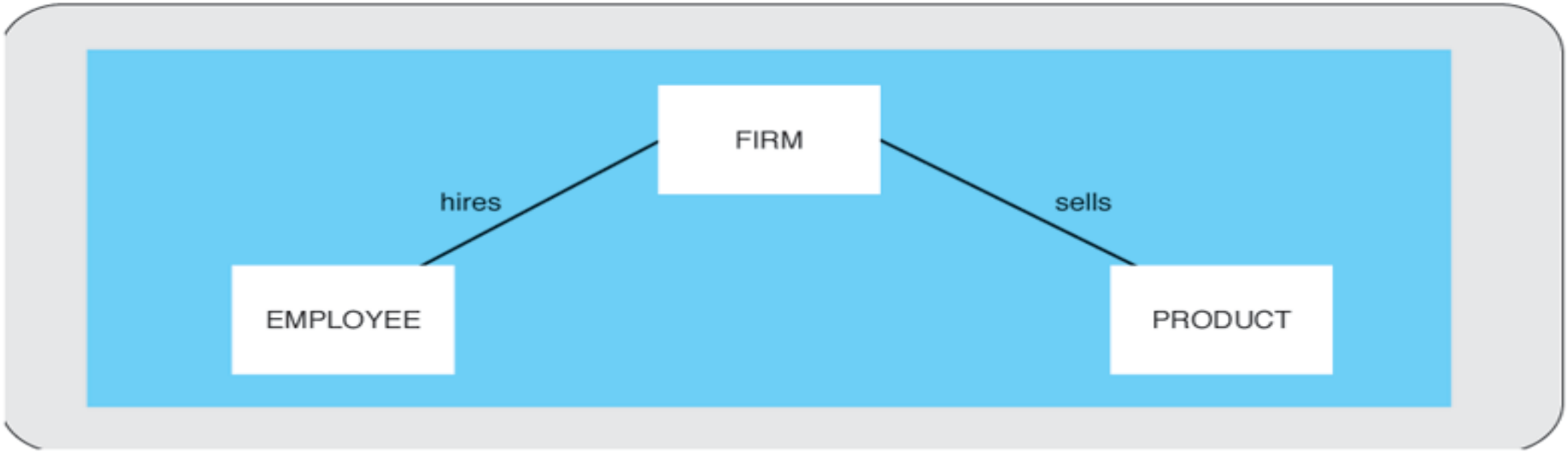


Figure 6.11 Entity-Relationship Diagram

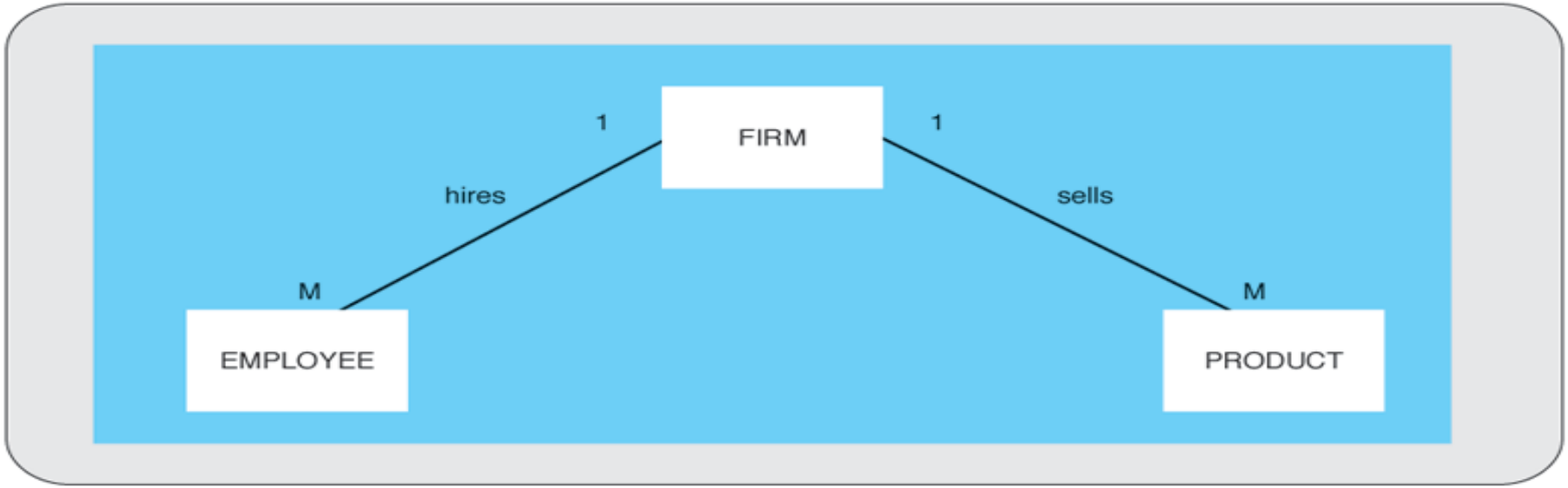
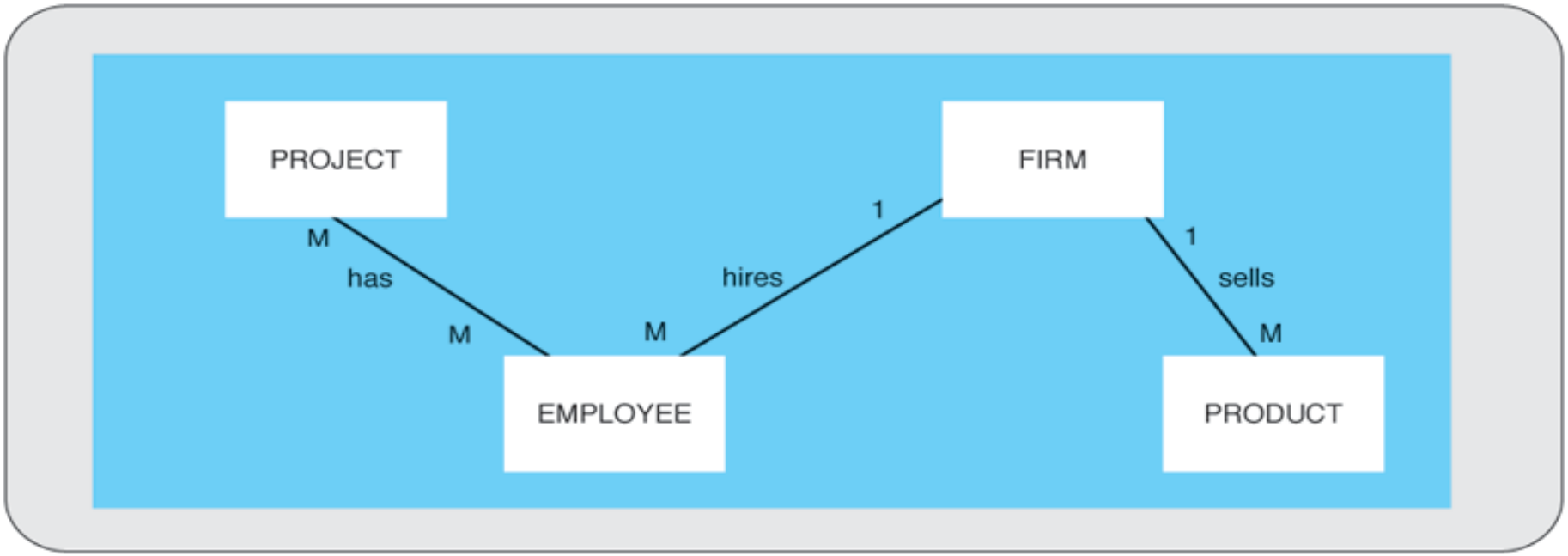
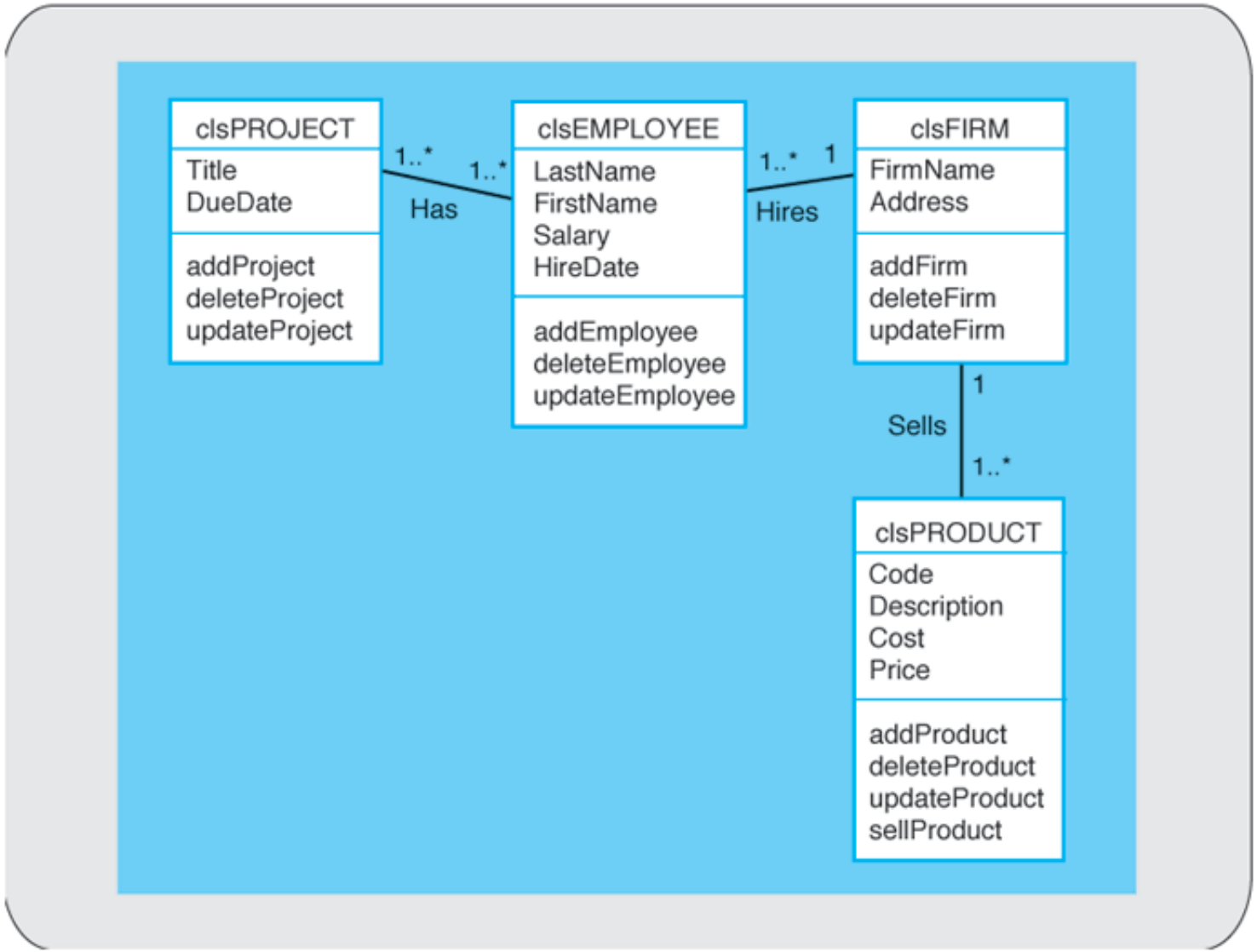


Figure 6.12 Entity-Relationship Diagram with a Many-to-Many Relationship



- When both the data used in an application and the actions associated with the data can be graphically represented they are called class diagrams and they are one of several object-oriented design models
- **Class diagrams** consist of the named class, fields in the class, and actions (sometimes referred to as methods) that act upon the class
- The class diagram in Figure 6.13 illustrates the entity-relationship diagram we have just completed

Figure 6.13 Class Diagram



USING THE DATABASE

- Consider a database on a personal computer
- Forms, reports, and queries are common methods for accessing the database held in a database management system
- A query language is the means for asking questions of the database
- Many database management systems provide an easy-to-use interface for the user

- The majority of users' interactions with databases are via reports and forms
- Graphical user interfaces (GUIs) are provided by most database management software vendors to make the development of forms and reports easier
- The greatest difference between forms and reports is in their format
- Figure 6.14 shows a form for entering courses into the database
- Figure 6.15 illustrates a form and subform combination

Figure 6.14 A Data Entry Form for the Course Table

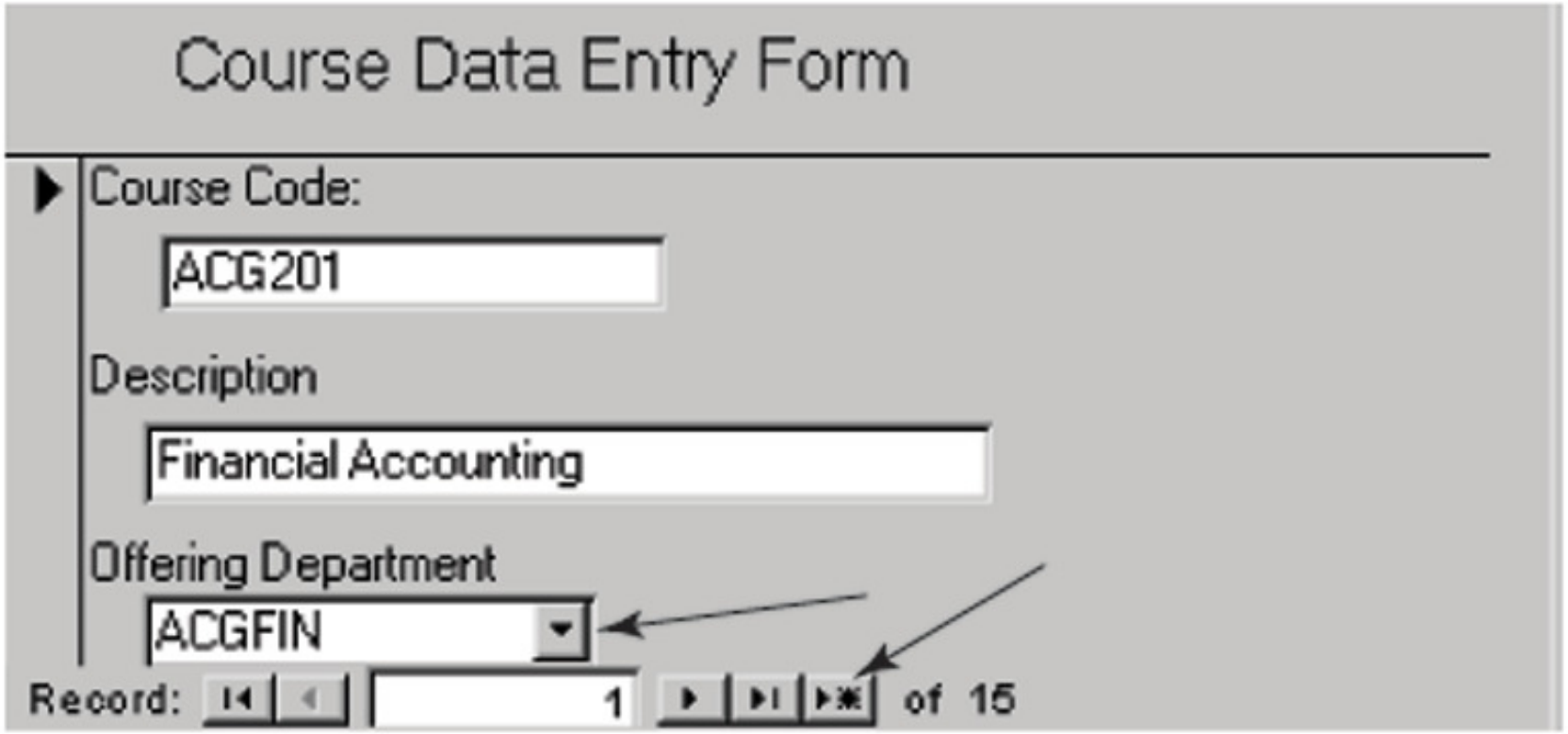
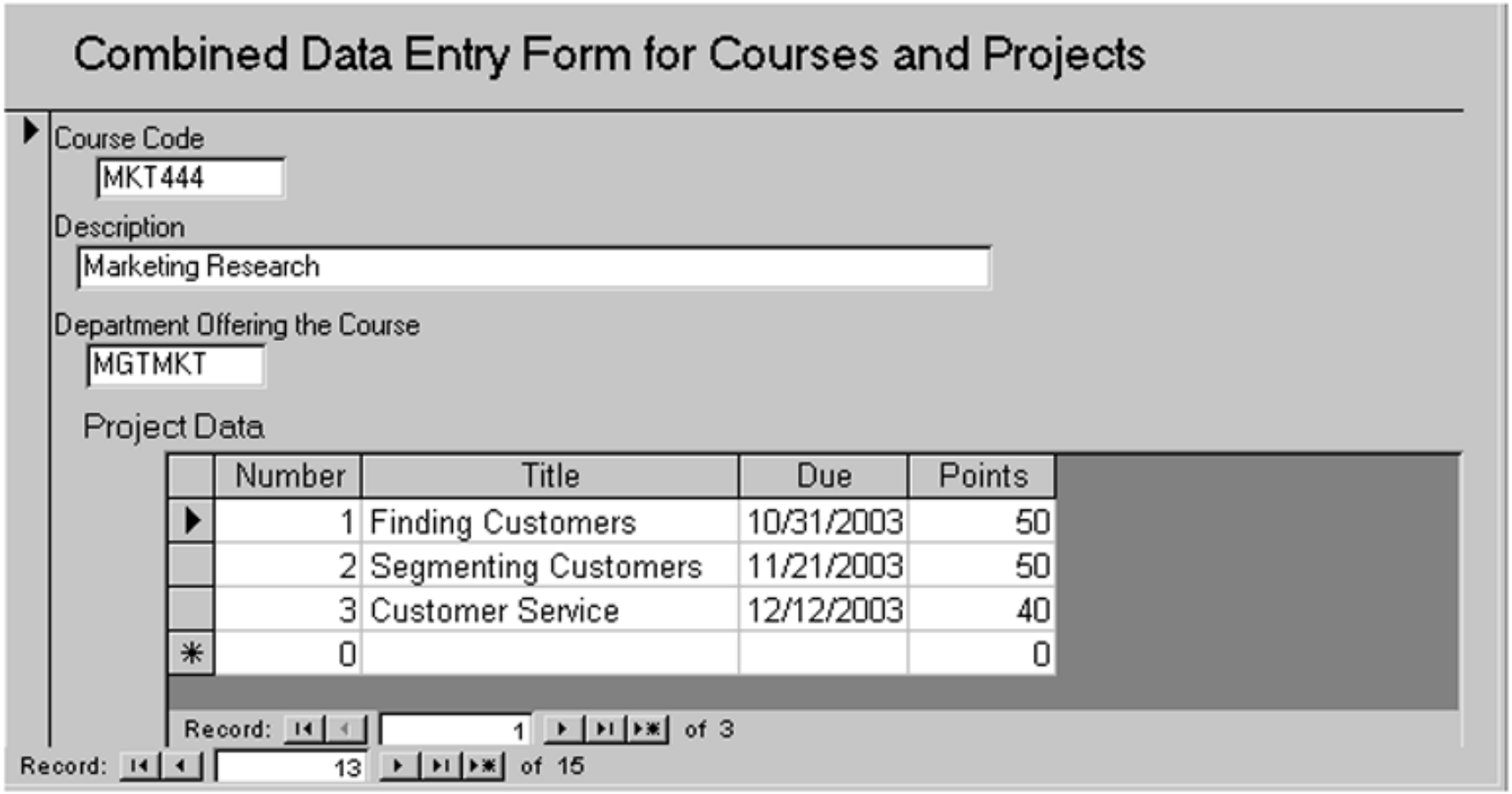


Figure 6.15 Combined Data Entry Form for the COURSE and PROJECT Tables



- **Reports** are aggregated database data formatted in a manner that aids decision making
- Figure 6.16 is a report that shows each department with a list of each course taught and projects required for the course
- Figure 6.17 illustrates that the DEPARTMENT table relates down to the COURSE table which, in turn, relates down to the PROJECT table
- Unless there was a related entry in the PROJECT table, no COURSE record was displayed. If no record from the COURSE table was used (for example, neither economics course had a project) then a DEPARTMENT record was not displayed

Figure 6.16 Report of Departments Showing Courses Offered and Course Projects

Courses by Department -- show projects

Department				
Accounting and Finance				
FIN305	*Personal Finance*			
		Project	*Due Date*	*Maximum Points*
	1	Personal Portfolio	11/14/2003	35
International Business				
INT201	*Spanish for Business*			
		Project	*Due Date*	*Maximum Points*
	1	Nouns	9/17/2003	15
	2	Verbs	11/21/2003	25
INT202	*French for Business*			
		Project	*Due Date*	*Maximum Points*
	1	Nouns	9/17/2003	15
	2	Verbs	11/21/2003	25
Information Systems and Operations Management				
MIS10	*Information Systems Literacy*			
		Project	*Due Date*	*Maximum Points*
	1	Home Page Development	9/15/2003	25
	2	Working With Windows	11/13/2003	50
MIS31	*Database Management Systems*			
		Project	*Due Date*	*Maximum Points*
	1	Alumni Database	12/5/2003	20
Management and Marketing				
MKT44	*Marketing Research*			
		Project	*Due Date*	*Maximum Points*
	1	Finding Customers	10/31/2003	50
	2	Segmenting Customers	11/21/2003	50
	3	Customer Service	12/12/2003	40

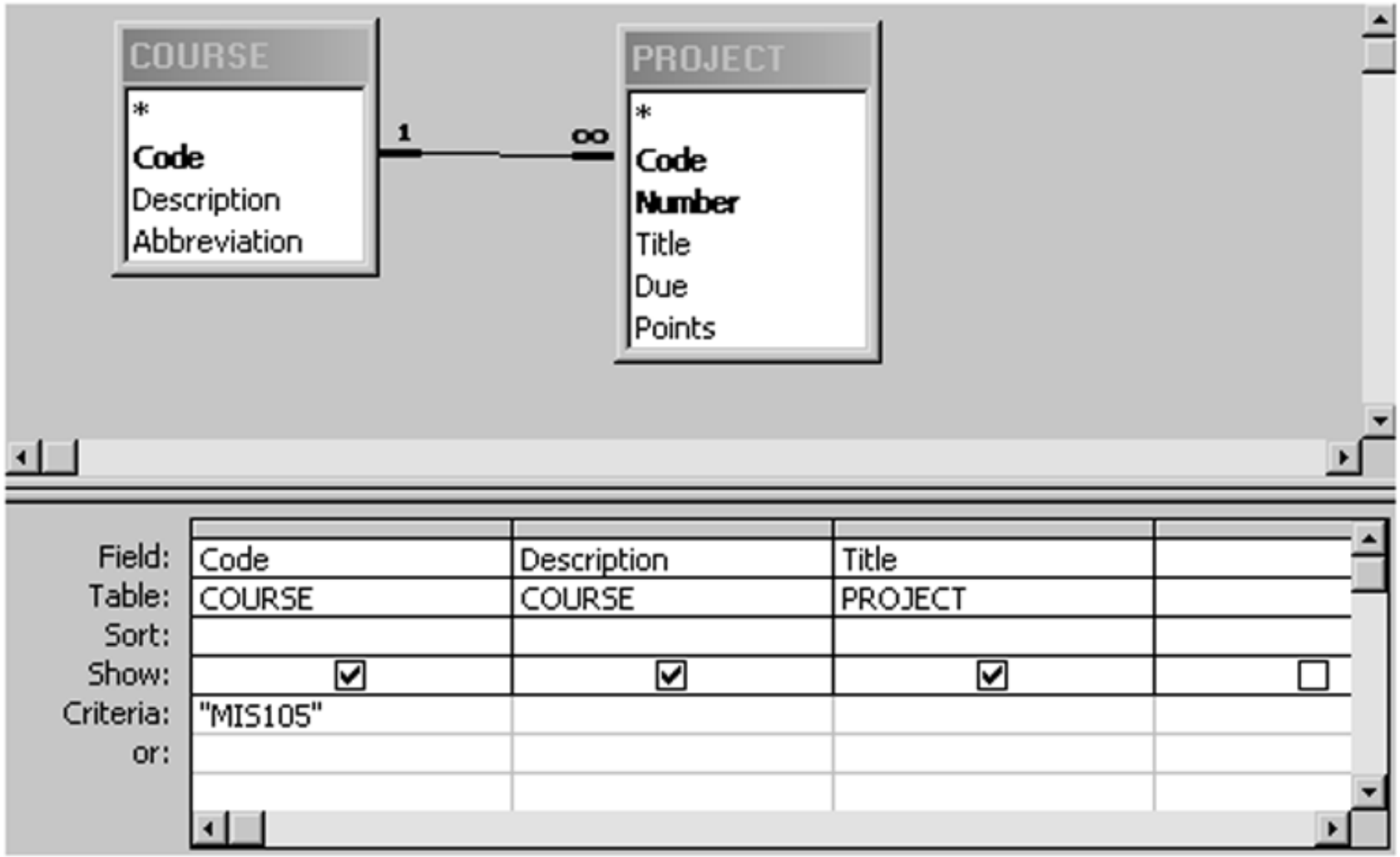
Figure 6.17 Report of Departments and Courses Alone

Courses by Department -- no projects

Department		
Accounting and Finance		
	ACG201	Financial Accounting
	ACG301	Cost Accounting
	FIN305	Personal Finance
Economics		
	ECN375	Global Markets
	ECN460	Banking Regulations
International Business		
	INT100	Cultural Diversity
	INT201	Spanish for Business
	INT202	French for Business
Information Systems and Operations Management		
	MIS105	Information Systems Literacy
	MIS315	Database Management Systems
	POM250	Introduction to Operations Management
	STA230	Descriptive Statistics
Management and Marketing		
	MGT300	Introduction to Management
	MKT300	Introduction to Marketing
	MKT444	Marketing Research

- Some users wish to go beyond reports and forms to directly ask questions of the database
- A **query** is a request for the database to display selected records and generally selects a limited number of data fields, then constrains the records to a set of criteria
- Figure 6.18 represents how that query could be represented
- The format is called **query-by-example (QBE)** because the DBMS software presents a standardized form that the user completes so that the system can generate a true query
- The result of the query is the table in Figure 6.19

Figure 6.18 Report of Departments and Courses Alone



- **Structured query language (SQL)** is the code that relational database management systems use to perform their database tasks
- While the user may see Figure 6.18 as the QBE, this is actually translated by database management system into the structured query language example shown in Figure 6.20
- DBMS software contains graphical user interfaces and "wizard" programs to walk users through queries in a user friendly manner
- **Online analytical processing (OLAP)** is another feature becoming more common in database management system software

Figure 6.19 Results of the Query-by-Example

	Code	Description	Title
▶	MIS105	Information Systems Literacy	Home Page Development
	MIS105	Information Systems Literacy	Working With Windows
*			

Figure 6.20 Structured Query Language Code to Find Projects for the MIS105 Course

Structured Query Language Code to Find Projects for the MIS105 Course

```
SELECT COURSE.Code, COURSE. Description, PROJECT.Title
FROM COURSE, PROJECT
WHERE COURSE.Code = PROJECT.Code
    AND COURSE.Code = "MIS105"
```

MANAGING THE DATABASE

- Database management systems perform functions that most users never see
- The infrastructure is needed so that the database can be maintained and modified and also to assure its efficient operation

- The **performance statistics processor** component of the DBMS maintains information that identifies what data is being used, who is using it, when it is being used, and so forth
- As the database management system runs, it keeps a **transaction log** that notes every database action taken as well as the exact time the action was taken
- A **backup** copy of the database is also made periodically

- The **database administrator (DBA)** has both technical and managerial responsibilities over the database resource.
- **Database programmers** create the databse applications required by firms for their corporate use
- The **database end-user**, by virtue of the decisions made and the amount of data retrieved, also has a major impact on database design, use, and efficiency

DATABASE MANAGEMENT SYSTEMS IN PERSPECTIVE

- The DBMS makes it possible to create a database, maintain its contents, and disseminate the data to a wide audience of users without costly computer programming
- Its ease of use allows managers and professional staff to access database contents with only modest training
- Every facet of information technology has both advantages and disadvantages and database management systems are no exception

- The DBMS enables both firms and individual users to:
  - Reduce data redundancy
  - Achieve data independence
  - Retrieve data and information rapidly
  - Improve security
- A decision to use a DBMS commits a firm or user to:
  - Obtain expensive software
  - Obtain a large hardware configuration
  - Hire and maintain a DBA staff

END OF CHAPTER 6