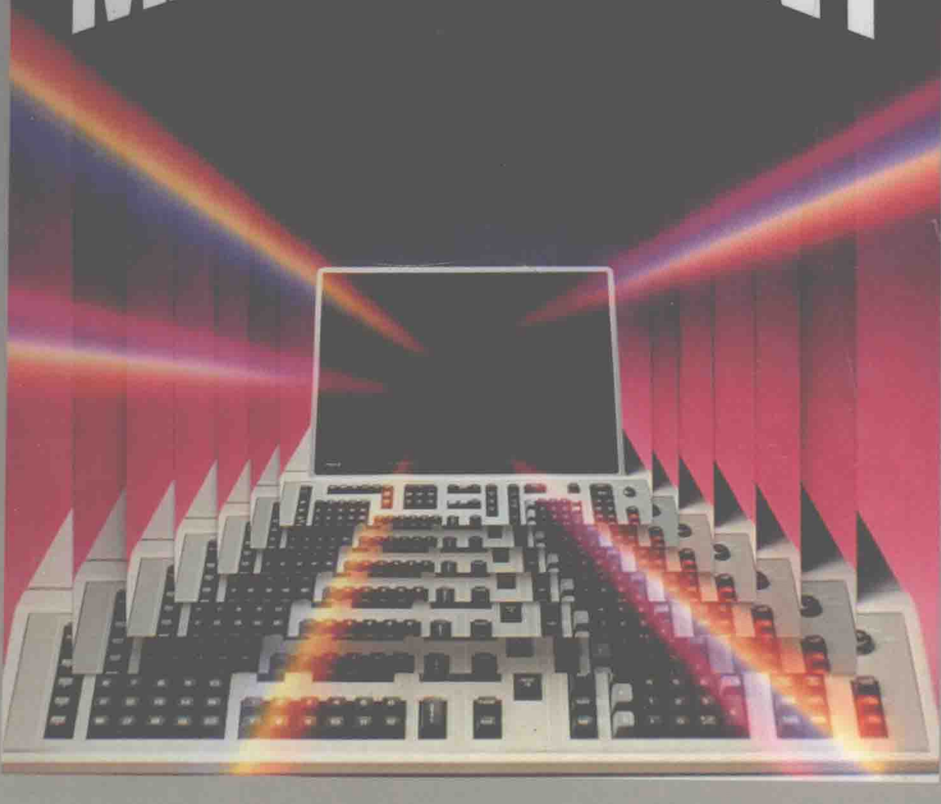
PRACTICAL DATABASE MANAGEMENT



Anthony J. Fabbri ■ A. Robert Schwab

PRACTICAL DATABASE MANAGEMENT

TONY FABBRI
University of Louisville

A. ROBERT SCHWAB, JR.
Provident Life Insurance

PWS-KENT PUBLISHING COMPANY
Boston

PWS-KENT
Publishing Company

20 Park Plaza
Boston, Massachusetts 02116

Sponsoring Editor: Jonathan Plant
Assistant Editor: Mary Thomas
Production Editors: Chris Crochetiere/Kirby Lozyniak
Manufacturing Coordinator: Lisa Flanagan
Production and Interior Design: Editorial Services of New England, Inc.
Cover Designer: Kirby Lozyniak
Typesetter: Grafacon, Inc.
Printer and Binder: Arcata Graphics/Halliday
Cover Printer: New England Book Components
Cover Image: Ellen Schuster. Courtesy of The Image Bank

PWS-KENT Publishing Company is a division of Wadsworth, Inc.

Library of Congress Cataloging-in-Publication Data

Fabbri, Tony.
Practical database management / Tony Fabbri and A. Robert Schwab, Jr.
p. cm.
Includes index.
ISBN 0-534-92592-8
1. Data base management. I. Schwab, A. Robert. II. Title.
QA76.9.D3F22 1992 91–39170
005.74—dc20 CIP

Printed in the United States of America
92 93 94 95 96—10 9 8 7 6 5 4 3 2 1

Preface

Database textbooks come in many varieties and all of them contain an abundance of information. Many of the textbooks are designed strictly for the business student, while others are designed for the computer scientist. This text is designed for both the business student who wants practical usage of database concepts, and the computer science major who needs to become exposed to the business side of data processing. Throughout the text, we include state-of-the-art database concepts as they apply to the business environment. Database theory throughout the text includes practical applications of that theory as it applies to the business world.

We wrote the text keeping in mind that the student reader will someday be using this information in his or her job function. We try not to include theory that has no practical usage in the real world. Because none of us knows what the future will create, however, we must include concepts and principles that look to the future of databases. Most computer facilities provide services to the business aspect of the corporation. Therefore, the concepts and principles in this text work for both the business major and the computer science major. The business major needs exposure to database concepts if he or she is to survive in today's business world, and the computer science major needs exposure to the underlying principles and needs of the business community if he or she plans to effectively function in the corporate world.

It is our belief that relational databases will dominate the database environment for the next few decades. This does not mean we can abandon the other types of databases. Remember, in the commercial world things are done based on the bottom line. Even though relational databases are the current pet of the computer industry, most industrial data centers will convert to them slowly—ever so slowly. Because most computer shops have substantial dollar investments in network and hierarchical databases, they must approach relational databases as they would any other new concept or product. They will perform a feasibility study, an in-house test, and then slowly build a library of relational programs and systems while maintaining their existing investment in older database structures.

Apple Corporation introduced its first "home" computer in the late 1970s; but personal computers were not prevalent in the business world until the late 1980s. And even now, many PCs are not used to their full potential. This span of over a decade reflects the business world's caution toward new technology and its inability to absorb any new technology quickly. A similar scenario will occur for relational databases. In our treatment of database concepts, we must include topics dealing with all three types of database structures: relational, hierarchical, and network. Because we feel relational databases will be the logical choice of most computer shops, we slant this textbook toward the relational model and devote only a few chapters to the other two database models.

Because many students are computer neophytes, the first three chapters of the text provide an introduction to computers. These chapters introduce the concepts students will need for understanding the remaining chapters of the text. If the reader already knows an idea or principle we discuss, however, he or she can ignore that section of the text and proceed to the next topic. Likewise, if the reader already has an excellent background in the mainframe and microcomputer environment, he or she can skip the first three chapters. These chapters provide an overview of files, databases, the mainframe and micro environments, and data structures. Students with only an elementary knowledge of computers should read Chapters 1, 2, and 3 more thoroughly.

Databases can be resident on a mainframe, minicomputer, or a microcomputer or some combination of the three. Most business environments include one or more mainframes to handle the bulk of the company's data processing needs and a battery of PCs to handle other data processing demands of the organization. The text includes detailed treatment of databases as they appear in both the mainframe and PC environments. The vast majority of computer shops are IBM mainframe oriented and most business organizations have hundreds if not thousands of IBM PCs and their compatibles. Because most business majors and information science majors will be on an IBM mainframe and an IBM PC (or compatible) when they reach the business world, this text leans heavily to these two computer environments.

We chose to concentrate on IBM's version of SQL throughout the text's treatment of relational databases. We did this for two reasons. First, IBM's treatment of SQL is becoming a de facto standard for SQL. Second, IBM's consortium agreement with most colleges and universities permits all colleges with IBM minis and mainframes to get SQL/DS virtually free. Because SQL/DS runs on many of IBM's minis (e.g., the AS 400 and the 9370), many colleges and universities can have access to SQL/DS for the price of a minicomputer. Although this is not a small sum of money, it still is within the budgetary framework of most colleges and universities.

After each chapter we include questions for the student to answer. In addition, many chapters include one or more case studies for the student to work. These studies are directed toward the PC environment and the mainframe environment. The commercial world uses both these environments, so the student should be exposed to both environments in the classroom. Remember, some solutions to real problems might exist only in the vastness of the mainframe setting, while other solutions might be ideally suited to the PC world. Students, therefore, need the tools and knowledge of both cultures to effectively function in today's computerized business world.

In some parts of the text we include COBOL examples and record formats to illustrate a concept or to define a record. Appendix A gives a brief summary of file description formats using the COBOL language.

In addition to the case studies and the problem exercises, we have included several minilabs for the student to work. We feel it is important for the student to have hands-on experience and the minilabs will give them some practical experience using database concepts.

Tony Fabbri
A. Robert Schwab, Jr.

Contents

Preface **xi**

CHAPTER 1 OVERVIEW OF FILES AND DATABASES 1

Traditional Files **1**
Bits and Bytes **2** Fields, Records, and Files **5**
Databases **8**
Fixed-length and Variable-length Records **8**
File Types **10**
Master Files **11** Transaction Files **1** History Files **12**
Backup Files **13** Input, Output, Print, and Sort Files **13**
File Accessing Methods **14**
File Storage and Access **14** Hashing **15**
Database Concepts **17**
The Complete Database Environment **17**
Procedures and Standards **17** Security Criteria **18** Training **18**
Computer Programs **18** Error-free Data **18**
Advantages of Databases **19**
Reduced Data Redundancy **19** Data Integrity **20** Data Independence **20**
Data Security **21** Data Consistency **21** Easier Use of Data **22**
Less Storage **24**
Disadvantages of Databases **24**
Complexity **24** Expense **24** Vulnerability **25** Size **26** Training Costs **27** Compatibility **28** Locking In of Technology **28** Lack of Lower-level Control **29**
Programmer and Programming Considerations **29**
Database Structures **30**
Entity Relationships **30**
Structures to Express Entity Relationships **31**
Hierarchical Databases **31** Network Databases **34** Relational Databases **36**

Summary **37**
Questions **39**
Case Study **39**

CHAPTER 2 HARDWARE AND OPERATING SYSTEM CONSIDERATIONS 41

The Mainframe Environment **41**
Operating Systems and Software Components **41** Overview of Task, Job, and Data Management **41** Layers of Software **42** Multiprogramming **43** Data Sets, Partitioned Data Sets, and Libraries **44** Systems Software **44**
Advanced Topic: Types of Operating Systems **46**
Data Storage on Mainframes **48** Extended Memory **50** Batch and On-Line Processing **61** Major On-Line Systems **63**
The Microcomputer Environment **66**
Hardware **66** Operating Systems **69** Local Area Networks **71** Real Storage on Microcomputers **72** External Storage on Microcomputers **74** Program Processing on Microcomputers **77**
Advantages and Disadvantages of Mainframes and Microcomputers **81**
Hybrid Systems **82**
Summary **84**
Questions **86**

CHAPTER 3 DATA STRUCTURES AND ACCESS METHODS 88

Physical Data Structures **88**
Data Types **88** Storage of Data **88** Integer Data Storage on Microcomputers **90** Packed and Unpacked Data Storage on Mainframes **91** Alphanumeric Data **92**
Advanced Topic: More on Alphanumeric Data **93**
Real Data **93**
Arrays **95**
Fixed-length and Variable-length Records **98**
Variable-length Fields **98** Variable-length Records **100**
Record Incompatibility **101**
Disk Labels **103**
File Structures **104**
Linked Lists **104** Trees **109**
Files and Access Methods **112**
VTOC **112** Blocked and Unblocked Files **115** Fixed versus Variable Blocking **115** Sequential Files and Logical Merging **118** Keys in Files **118**
Batch Processing of Database Records **119**
Hashing and Direct Access Files **120** Alternate Randomizations **121**
Randomizing Methods **122**
Remainder Method **122** Folding Method **122**
Overflows **123**
Numeric Limits **123** Limits Using Hardware **123** Prime Blocks **124**

Indexed Files 126
Sequential Indexes 126 Sorted Indexes and Binary Searches 126 High-level Indexes 129 Tree Formats for Indexes 131
VSAM 133
Kinds of VSAM Data Sets 134
Summary 134
Questions 136
Case Study 137

CHAPTER 4 DATABASE MANAGEMENT SYSTEMS 138

DBMS Components and Functions 138
The Data Interface 138 The User Interface 138 The Programmer Interface 139
Using a Database Management System 139
The Database Designer 139 The Database Programmer 143
The Database Administrator 146
Managing the Database 146 Providing Security for the Database 147 Monitoring System Performance 147 Planning Backup and Recovery 147 Maintaining Skills 147
The End User 148
Role in System Development and Database Design 148 Role in Accessing the Database 149
Summary 149
Questions 150

CHAPTER 5 DBMS CONCEPTS AND ARCHITECTURE 151

Accessing Records on the Database 151
Using an Index to Store Records 152 Using a Hashing Routine to Store Pages 154 Makeup of a Database Page 154 Keys and Indexes 155
Number of Records Accessed Considerations 156
Types of Relationships 156 Coding Logic in One-to-One and One-to-Many Relationships 158
Using Foreign Keys 160
Accessing a One-to-Many Relationship Using Indexes 160 Many-to-Many Relationships 162
Data Models 164
Hierarchical Models 164 Network Models 165 Relational Models 169 Disadvantages of Network and Hierarchical Designs 171 Advantages of Relational Designs 172
Schema, Subschema, and DMCL 172
The Schema and Subschema 172 The DMCL 173 Load Libraries 175 The Subschema 175
DBMS Catalogs and Their Data Dictionaries 177
Database Languages 178
DL/I 178 Structured Query Language 180

The Database Environment 182
The DB2 and SQL/DS Environment 182 The IDMS/R Environment 182
The IMS Environment 183 The PC Environment 183
Summary 184
Questions 185
Case Study 186

CHAPTER 6 CONCEPTUAL DATABASE DESIGN 188

Overview of System Development 188
A System Development Methodology 191
Relationships in Conceptual Databases 195
Parent-Children versus Owner-Members versus Tables 196
Database Design Tools 198
Data Flow Diagrams 198 Data Dictionaries 200 Data Hierarchy Charts 201 Entity-Relation Diagrams 201 CASE Tools 202
Designing a Relational Database 203
Defining the Data Entities of the Database 207 Defining the Relationships between Data Entities 208 Translating the Relationships into Tables 212
Summary 216
Questions 216
Case Study 217

CHAPTER 7 NORMALIZATION 219

Overview of Normalization: Theory versus Practice 219
The Normal Forms 223
Normalization as a Design Methodology 225
Fourth and Fifth Normal Forms 229
Summary 233
Questions 233
Case Study: Normalization Minilab 234

CHAPTER 8 PHYSICAL DATABASE DESIGN 237

Trade-offs between Minimal Data Redundancy and Performance 237
Referential Integrity 240
Converting Conceptual Database Design to Physical Database Design 241
Defining All Columns of All Tables 241 Determining Whether (and Where) to Override Normalization 242 Defining Referential Integrity Requirements 243 Defining Essential Views 243 Defining Essential Indexes 245
Summary 245
Questions 246
Case Study 246

CHAPTER 9 RELATIONAL DATABASES AND SQL 247

The Relational Model 247
Using Data Definition Language 248
Creating Entities 248 Creating Databases and Tablespaces 248 Creating Tables 250 Defining Columns 251 Creating Unique Indexes 252 DASD Usage by User Departments 252 Creating Nonunique Indexes 254
Using Data Manipulation Language 255
The SELECT Statement 256 The INSERT Statement 271 Using Synonyms and Views 273 The DELETE Statement 274 The UPDATE Statement 275 Commercial Usage 276 Joining Multiple Tables 276 Subselects 278
Control Language 281
The GRANT Keyword 281 The REVOKE Keyword 283 Automatic Authorizations 285
Summary 285
Questions 288
Special Project: Minilab Assignment 290
Case Study 291

CHAPTER 10 RELATIONAL DATABASE IMPLEMENTATIONS 292

Referential Integrity 292
Parent and Dependent Tables 293
Parent Table Rules 293 Dependent Table Rules 294 Initiating Referential Integrity 295
Implementing Referential Integrity Using DDL 295
Using the CREATE Statement 295 Using the ALTER Statement 298
Eliminating Referential Integrity from Tables 299
Loading Data into a Relational Database 299 Using the LOAD Utility 300
Physical Structure and Characteristics 302
A Look at DB2 302 Tablespace Structure 303 Types of Tablespaces 304
The DB2 Catalog 310
Accessing the DB2 Catalog 310 Accessing Specific DB2 Catalog Tables 311
Systems Interfacing 314
DB2 and TSO 314 DB2 and IMS or CICS 314
Summary 315
Questions 316

CHAPTER 11 MICROCOMPUTER RELATIONAL DATABASE IMPLEMENTATIONS 318

dBASE Overview 318
The dBASE End-User Interface 319
ASSIST in dBASE III PLUS 319

Creating an Entity **320**
Creating a Database **320** Creating the Field Description for the Database **320**
Using the Database **321**
Placing Data in the Database **321** Manipulating Data in the Database **322** Deleting and Recalling a Record **322** Creating Indexes **322**
The dBASE IV Control Center **322**
dBASE Programming **324**
A Simple Application **324**
The Application's Design **324**
Option A—The Employee Database Selection **324** Option B—The Office Database Selection **327** Option C—List Phone Directory Selection **327**
Writing Code in dBASE **327**
Using dBASE's Program Editor **328**
A Description of the Employee Main Menu Program **328**
Using Comments and Documenting the Program **328** The SET Command **328** The STORE Command **330** The DO WHILE Command **330** The CLEAR, PRINT, SAY, and WAIT Commands **331** Processing the User's Choice **332** The DO CASE Command **332** The RETURN and USE Commands **333** A Description of the Employee Submenu Program **333** The SET PROCEDURE Command **333** The SET FORMAT and APPEND Commands **334** Some Logical Considerations **335** Using a Procedure **336** Identifying Parameters **337** The @ . . . Commands **337** The GOTO and LOCATE Commands **337** The IF Command **338** Handling Error Messages **338** The @ . . . CLEAR Command **339** Concatenating Alphanumeric Strings **339** The WAIT Command **339** Code for Changing a Record **340** Using Data from Two Databases **340** The SELECT and USE Commands **341** The SET RELATION Command **341** The LIST Command **342**
dBASE IV and SQL **342**
Accessing SQL via the SET SQL Command **343**
Summary **343**
Questions **344**

CHAPTER 12 END USERS AND RELATIONAL DATABASES 345

Accessing Databases **345**
DB2 and Its Associated Products **346**
DB2I **347**
QMF **347**
Overview **347** End Users and QMF **348** Using QMF **350** The Query Screen **351**
Using and Saving Queries and Forms **352**
Using Queries, Data, and Forms **352** Starting Over **354** Saving a Query, Its Data, and Form **354** Deleting Presaved Queries, Forms, and Data **354** Naming Conventions of Saved Items **355** Error Messages **355**

The Report Screen 356
The Forms Screen 357
The Main Forms Screen 357 Displaying Headings on Multiple Rows 358 Edit and Width Codes 358 Usage Codes 360 Other Screens for Formatting the Report 361 The Page Subscreen 362 Performing Control Breaks 365
Procedures 366
Summary 366
Questions 368
Case Study 369
Minilab 369

CHAPTER 13 SQL AND THE PROGRAMMER 372

COBOL and Embedded SQL 372
Program Preparation 372 The Precompiler 373 Delimiters for Embedded SQL 374
The Bind and Plan 375
Authorizations 378
Time Stamps 379
Embedded SQL 380
Working-Storage Considerations 381 The DCLGEN 381 The DECLARE Statement 382 Using Host Variables 384 Using Cursors 386 Updating Using a Cursor 391 Saving Work 393 Deleting Using a Cursor 395 Inserting Rows via a Program 396
Execution of Embedded-SQL COBOL Programs 399
DB2I 400
Converting Nonrelational Files to Relational Tables 402
Converting Microcomputer Databases to Mainframe Databases 402
Converting Mainframe Databases to Microcomputer Databases 403
Summary 403
Questions 405
Case Study: A Comprehensive Lab Exercise 405

CHAPTER 14 NETWORK DATABASES 408

Network Structure 408
Occurrence Diagrams 409
The Schema 412
The Set Definition 413
The Subschema 418
A Look at IDMS/R 419
Designing an Application Using IDMS/R 420 OLM 420 ADS/A and ADS/G 421 IDD 422
Summary 422
Questions 423

CHAPTER 15 HIERARCHICAL DATABASES **424**

Overview of IMS **425**
Basic Definitions for Hierarchical Databases **425**
The Hierarchical Model **425**
The IMS Database **430**
The Logical Database **430** The Secondary Index **431** Schema and Subschema: The DBDGEN and the PSBGEN **432**
Physical Database Structure **438**
HSAM Databases **439** HISAM Databases **441** HDAM Databases **442** HIDAM Databases **446**
Manipulating the Database Using DL/I **447**
Elements of DL/I **447** Command Codes **450** Writing IMS Programs in COBOL **452**
Executing IMS Programs **452**
Coding DL/I Batch Programs **452** Coding MPPs **453** Coding BMPs **456**
Designing a Hierarchical Database **456**
Summary **457**
Questions **458**
Case Study **458**

CHAPTER 16 SPECIAL TOPICS IN DATABASES **461**

Concurrent Database Processing **461**
Distributed Databases **462**
Artificial Intelligence and Databases **466**
Object-oriented Programming and Databases **466**
Hypertext **467**
Summary **469**
Questions **470**

APPENDIX A SHORT DISCUSSION ON DEFINING COBOL VARIABLES **471**

The Data Division **471**

APPENDIX B USING DL/I IN AN APPLICATION PROGRAM **474**

APPENDIX C INTRODUCTION TO SQL/DS **482**

Running Queries in Batch Mode **483**
Running Queries Online **483**

APPENDIX D ACRONYMS AND THEIR MEANINGS **484**

Index **490**

1 *Overview of Files and Databases*

To understand database management, you must first develop a sense of the context in which databases are used. The purpose of these first three chapters is to present three components of that context. This chapter examines the similarities and differences between databases and traditional (nondatabase) files. Chapter 2 examines the machine environment by focusing on hardware and its operating system software, because their combination determines the way in which a database can be used. Chapter 3 examines the data structures and data access methods that are used by databases.

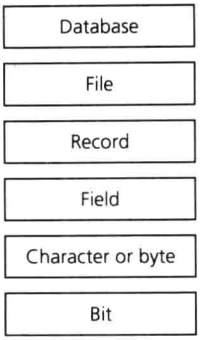
This chapter begins by examining traditional files, because they are the foundation of databases. We then examine the advantages and disadvantages of using databases instead of traditional files. We examine briefly the three major types of databases that have evolved for the storing of data.

TRADITIONAL FILES

In the typical business environment the computer provides a variety of reports to a company's management. A few of these include the following:

1. Reports on the financial status of the company
2. Reports on the company's customers, clients, and competitors
3. Reports on the operational status of the company and its employees
4. Reports providing documentation to various government regulators

In each of these cases the computer converts data into usable information by processing the data using a computer program. Data can be divided into several levels, called a **data storage hierarchy** (see illustration on page 2). The bit is the smallest and simplest entity in the hierarchy, and the database is the most complex. The discussion starts with the simplest entity—a bit.



Bits and Bytes

A **bit,** or binary digit, is a single-digit number represented by either 0 or 1, and it is the smallest unit of storage in the computer. A **character,** or **byte,** is a combination of eight consecutive bits. A character represents one storage location in the computer where we can store either **alphabetic characters** (*A* through *Z*, and *a* through *z*), **numeric characters** (0 through 9), and **special characters** (+, %, $, and so forth). An **alphanumeric character** is any alphabetic, numeric, or special character. Because the smallest bit configuration for a character is 0000 0000 and the largest bit configuration is 1111 1111, we can store any one of a possible 256 different symbols in one character or byte. (Note that binary numbers usually are written in groups of four: The binary number for 25 (decimal) is 11001; this figure is written as 0001 1001. This is done for purposes of readability, just as we typically separate decimal numbers into groups of three, separated by commas: for example, 195,674,890.)

We convert the 256 possible 8-bit codes into a character using either the Extended Binary Coded Decimal Interchange Code (EBCDIC) or American Standard Code for Information Interchange (ASCII) encoding standards (see Figure 1.1).

Typically, large IBM mainframes use the EBCDIC coding standard, and minicomputers and microcomputers use the ASCII coding standard. The bit configuration for the character A in EBCDIC (1100 0001) is different from the bit configuration for the A in ASCII (0100 0001). Consequently, when transferring data from one computer that uses ASCII to a computer that uses EBCDIC, (such as transferring data from a PC to a mainframe), you must "massage" the data before or after transferring it. This massaging of the data is done through a special program called a **utility program** (Figure 1.2). A utility program performs a special service or function that is needed by several applications. For example, it makes no difference whether you are transferring data for a billing system, a management report, or a budgeting system; you still must translate the data, and the same translation program can be used by each application. A special program or routine must transform each character stored in the ASCII coding system to a character in the EBCDIC coding system.

The capacity of primary and secondary memory in computer systems is measured in kilobytes (K), megabytes (M or meg), gigabytes (G or gig), or terabytes (T). A **kilobyte** is 1024 bytes or characters and a **megabyte** is 1024K, or 1,048,576 bytes or characters. On the other hand, a **gigabyte** is 1024M (about one billion) characters and a **terabyte** is 1024G (about one trillion). Hence, a 640K PC computer system with a 1.2-meg floppy disk drive and a 50-meg hard disk drive has 655,360 bytes of main memory, about 1,258,000 bytes available on a floppy diskette and about 50,000,000 bytes or locations available on the hard drive.

Parity Check Because machines have a tendency to break down and memory units can fail, many manufacturers include a parity check as part of the

Character	EBCDIC code		ASCII code		Character	EBCDIC code		ASCII code	
	Hex	Binary	Hex	Binary		Hex	Binary	Hex	Binary
A	C1	1100 0001	41	0100 0001	5	F5	1111 0101	35	0011 0100
B	C2	1100 0010	42	0100 0010	6	F6	1111 0110	36	0011 0110
C	C3	1100 0011	43	0100 0011	7	F7	1111 0111	37	0011 0111
D	C4	1100 0100	44	0100 0100	8	F8	1111 1000	38	0011 1000
E	C5	1100 0101	45	0100 0101	9	F9	1111 1001	39	0011 1101
F	C6	1100 0110	46	0100 0110	a	81	1000 0001	61	0110 0001
G	C7	1100 0111	47	0100 0111	b	82	1000 0010	62	0110 0010
H	C8	1100 1000	48	0100 1000	c	83	1000 0011	63	0110 0011
I	C9	1100 1001	49	0100 1001	d	84	1000 0100	64	0110 0100
J	D1	1101 0001	4A	0100 1010	e	85	1000 0101	65	0110 0101
K	D2	1101 0010	4B	0100 1011	f	86	1000 0110	66	0110 0110
L	D3	1101 0011	4C	0100 1100	g	87	1000 0111	67	0110 0111
M	D4	1101 0100	4D	0100 1101	h	88	1000 1000	68	0110 1000
N	D5	1101 0101	4E	0100 1110	i	89	1000 1001	69	0110 1001
O	D6	1101 0110	4F	0100 1111	j	91	1001 0001	6A	0110 1010
P	D7	1101 0111	50	0101 0000	k	92	1001 0010	6B	0110 1011
Q	D8	1101 1000	51	0101 0001	l	93	1001 0011	6C	0110 1100
R	D9	1101 1001	52	0101 0010	m	94	1001 0100	6D	0110 1101
S	E2	1110 0010	53	0101 0011	n	95	1001 0101	6E	0110 1110
T	E3	1110 0011	54	0101 0100	o	96	1001 0110	6F	0110 1111
U	E4	1110 0100	55	0101 0101	p	97	1001 0111	70	0111 0000
V	E5	1110 0101	56	0101 0110	q	98	1001 1000	71	0111 0001
W	E6	1110 0110	57	0101 0111	r	99	1001 1001	72	0111 0010
X	E7	1110 0111	58	0101 1000	s	A2	1010 0010	73	0111 0011
Y	E8	1110 1000	59	0101 1001	t	A3	1010 0011	74	0111 0100
Z	E9	1110 1001	5A	0101 1010	u	A4	1010 0100	75	0111 0101
0	F0	1111 0000	30	0011 0000	v	A5	1010 0101	76	0111 0110
1	F1	1111 0001	31	0011 0001	w	A6	1010 0110	77	0111 0111
2	F2	1111 0010	32	0011 0010	x	A7	1010 0111	78	0111 1000
3	F3	1111 0011	33	0011 0011	y	A8	1010 1000	79	0111 1001
4	F4	1111 0100	34	0011 0100	z	A9	1010 1001	7A	0111 1010

FIGURE 1.1 *The EBCDIC and ASCII encoding systems.*