

BUSINESS DATABASE TECHNOLOGY

BUSINESS DATABASE TECHNOLOGY

**An Integrative Approach to Data Resource Management with
Practical Project Guides, Presentation Slides, Answer Keys to
Hands-on Exercises for Students in Business Programs**

**SHOUHONG WANG
HAI WANG**



Universal-Publishers
Boca Raton

Business Database Technology: An Integrative Approach to Data Resource Management with Practical Project Guides, Presentation Slides, Answer Keys to Hands-on Exercises for Students in Business Programs

Copyright © 2014 Shouhong Wang and Hai Wang
All rights reserved.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher

Universal-Publishers
Boca Raton, Florida
USA • 2014

ISBN-10: 1-61233-288-9
ISBN-13: 978-1-61233-288-8

www.universal-publishers.com

Cover image © Can Stock Photo Inc./ kgtoh

Credit: ORACLE, MySQL are trademarks of Oracle Corporation. IBM DB2 is trademark of IBM. Windows, Microsoft SQL Server, Microsoft Office, Microsoft Access, Microsoft Excel, and Microsoft Visual Studio are trademarks of Microsoft Corporation.

Library of Congress Cataloging-in-Publication Data

Wang, Shouhong.

Business database technology : an integrative approach to data resource management with practical project guides, presentation slides, answer keys to hands-on exercises for students in business programs / Shouhong Wang and Hai Wang.

p. cm.

Includes index.

ISBN-13: 978-1-61233-288-8 (pbk. : alk. paper)

ISBN-10: 1-61233-288-9 (pbk. : alk. paper)

1. Business--Databases. 2. Information technology--Technological innovations. 3. Database management. 4. Database design. 5. Management information systems. I. Wang, Hai, 1973- II. Title.

HF5548.2.W2992 2013

005.74--dc23

2013039227

Table of Contents

PREFACE	7
CHAPTER 1. INTRODUCTION.....	11
1.1. Database Technology.....	11
1.2. Data Are Resource of the Organization	11
1.3. Data, Information, Knowledge	12
1.4. Common Mistakes in Data Resource Management.....	13
1.5. Control Data Redundancy	15
1.6. Database and Database System.....	17
1.7. Database Management Systems	18
1.8. Commonly Used DBMS for Relational Database.....	21
CHAPTER 2. DATA STRUCTURE TECHNIQUES FOR DATABASE.....	25
2.1. Data Secondary Storage - Disk.....	25
2.2. File, Record, Attribute, and Key	26
2.3. Pointer.....	27
2.4. Basic File Organizations.....	27
2.4.1. Sequential file.....	28
2.4.2. Random file.....	29
2.4.3. Indexed file	31
2.5. B-tree.....	33
2.5.1. Overview of B-tree	33
2.5.2. Construction of B-tree	34
2.5.3. B-tree maintenance	35
CHAPTER 3. DATA MODELS	43

Table of Contents

3.1. Overview of Data Models.....	43
3.2. ER Model	44
3.3. Entity, Attribute, and Primary Key.....	44
3.4. Relationship.....	46
3.5. Instrument for Implementing 1:1 and 1:M Relationships – Foreign Key	52
3.6. Instrument for Implementing M:M Relationships – Associative Entity.....	55
3.7. Summary of ER Diagram Convention.....	58
3.8. Construction of ER Diagram	60
3.8.1. Transcript	60
3.8.2. Sample datasheets	62
3.8.3. Redundant relationships in ERD.....	63
3.8.4. Iterations of ERD construction.....	65
CHAPTER 4. RELATIONAL DATABASE.....	69
4.1. Relational Data Model and Tables.....	69
4.2. Candidate Key and Alternative Key	70
4.3. Conversion of ER Model to Relational Data Model	71
4.4. Data Retrieval from Relational Database.....	76
4.5. Referential Integrity.....	79
CHAPTER 5. NORMALIZATION AND LOGICAL DATABASE DESIGN.....	85
5.1. Normalization	85
5.2. Functional Dependency	85
5.3. Normal Forms	86
5.3.1. Unnormalized form	87
5.3.2. Conversion from ONF to a normal form	88

Table of Contents

5.3.3. First Normal Form (1NF)	89
5.3.4. Data redundancy and data modification anomaly.....	90
5.3.5. Partial key dependency in 1NF table, and normalize 1NF	91
5.3.6. Second Normal Form (2NF) and non-key dependency.....	93
5.3.7. Normalize 2NF table with non-key dependency and Third Normal Form (3NF).....	93
5.3.8. Summary of normalization procedure from 0NF to 3NF.....	95
5.3.9. Boyce-Codd Normal Form (BCNF).....	97
5.3.10. Normalize 3NF table with reverse dependency into BCNF.....	99
5.3.11. Fourth Normal Form (4NF).....	100
5.3.12. Normalize BCNF table with multivalued dependency into 4NF.....	102
5.4. The Nature of Normalization and Higher-Level Normal Forms.....	103
5.5. Logical Database Design.....	103
CHAPTER 6. DATABASE PROCESSING AND SQL.....	113
6.1. Introduction to SQL.....	113
6.2. CREATE and DROP.....	114
6.3. INSERT, UPDATE, DELETE.....	115
6.4. Query - SELECT.....	117
6.5. WHERE Clause and Comparison.....	119
6.6. User Input Request.....	121
6.7. ORDER BY Clause.....	122
6.8. Aggregate Functions.....	123
6.9. GROUP BY Clause and HAVING Clause.....	124
6.10. Arithmetic Operations.....	125
6.11. Joining Tables.....	126

Table of Contents

6.12. Alternative Format of Inner Join and Outer Join	128
6.13. Subquery	130
6.13.1. Subquery - reducing computational workload of join operation.....	130
6.13.2. Subquery as an alternative to GROUP BY.....	132
6.13.3. Subquery - representing a variable.....	133
6.13.4. Subquery - determining an uncertain criterion.....	133
6.14. UNION Operator.....	136
6.15. Tactics for Writing Queries	136
6.16. SQL Embedded in Host Computer Programming Languages	137
CHAPTER 7. PHYSICAL DATABASE DESIGN.....	143
7.1. Physical Design.....	143
7.2. Adding Index	144
7.3. Adding Subschema.....	144
7.4. Clustering Tables	145
7.5. Merging Tables	146
7.6. Horizontal Partitioning Table.....	146
7.7. Vertical Partitioning Table	147
7.8. Creating New Primary Key.....	148
7.9. Substituting Foreign Key.....	148
7.10. Duplicating Table or Duplicating Part of Partitioned Table	149
7.11. Storing Information (Processed Data).....	149
7.12. Implementation of Physical Database Design.....	150
CHAPTER 8. DATABASE ADMINISTRATION	155
8.1. Data Planning and Database Design.....	155

Table of Contents

8.2. Data Coordination.....	155
8.3. Data Security, Access Policies, and Data Ownership	155
8.4. Data Quality	155
8.5. Database Performance.....	156
8.6. User Training.....	156
8.7. Data Standards, Data Dictionary, and Documentation.....	156
8.8. Database Backup and Recovery	158
8.9. Data Archiving.....	160
8.10. Database Maintenance.....	160
8.11. Managing Business Rules Related to the Database Design	161
CHAPTER 9. DATABASE IN COMPUTER NETWORKS	163
9.1. Centralized Database in the Local Area Network Environment	163
9.2. Centralized Database in the Internet Environment	163
9.3. Distributed Databases.....	166
9.4. XML for Databases.....	171
CHAPTER 10. DATA WAREHOUSE	177
10.1. Data Warehouse	177
10.2. Multidimensional Data and Data Cube.....	177
10.3. Creating Data Cube from Relational Database.....	178
10.4. Definition of OLAP	180
10.5. Analytical Processing Techniques of OLAP.....	181
10.6. Business Intelligence and Data Mining.....	183
10.7. Data Resource for Organizational Knowledge Development.....	184
TECHNICAL GUIDE A. CONSTRUCTING DATABASE USING MICROSOFT ACCESS	189

Table of Contents

TECHNICAL GUIDE B. AN EXAMPLE OF NORMALIZATION.....	231
TECHNICAL GUIDE C. OLAP USING MICROSOFT OFFICE	239
ANSWERS TO EXERCISE QUESTIONS AND REVIEWS.....	267
INDEX.....	284
POWERPOINT SLIDES.....	291

*** Electronic teaching material for this textbook includes model syllabus, answers to all assignment questions, sample exams, answers of the exams, lecture PPT, the Microsoft Access database for the textbook SQL examples, Microsoft Access database for Technical Guide A, Microsoft Access database for Technical Guide C, and others.*

PREFACE

Data are valuable assets of the organization. Database is a key component of the information systems in business. Database technology is critically important to business. This book is designed for business students who study database technology for data resource management.

The database management systems are powerful tools for constructing and accessing databases. Currently, many user-friendly end-user oriented relational database management systems are available on the commercial software market or are accessible as open source software products. However, people often do not fully realize the importance of understanding accurate concepts of database. In fact, a poorly designed database or misuse of database management systems would do more harm than good for the organization. The objective of this book is to help students understand the precise concepts of database and develop practical skills of database design, implementation, and application. Upon the completion of study of this book, students should be able to build and to manage databases in a professional way. More importantly, students should be able to develop their independent learning ability to learn advanced features of database design, implementation, application, and management.

IT has been the most innovative field in the modern era. Ironically, after it was first introduced to the computational world longer than three decades ago, the relational database model is still the most popular database model in the IT industry. Although research into object-oriented database has been carrying on for many years, the relational database model is still the main stream of databases in business. There is no evidence that the relational database model will phase-out any soon.

There have been hundreds textbooks of database on the market. Given the long history of database, many database textbooks were written decades ago and revised for numerous times. The volumes of these database textbooks are usually huge since they contain many secondary contents which might be useful decades ago but are no longer essential to know, especially for business students. On the other hand, contemporary topics, such as physical database design and database application for business intelligence (e.g., OLAP) are lacking in these textbooks. Furthermore, some textbooks are totally database management systems independent, but others seem to over-

emphasize specific database management systems (e.g., Oracle or Microsoft Access). This book maintains a good balance between the core concepts and secondary concepts, and includes both basic knowledge of database and hands-on material for Microsoft Access, a widely available end-user oriented relational database management system. In this book, a huge amount of material about database design, implementation, and application is boiled down to a practically workable volume.

The book is self-contained. It includes ten chapters of fundamentals of database technology, sufficient problem solving exercise questions for each chapter, answers to the representative exercise questions, key learning objectives of each chapter, three tutorials for projects of database technology applications or large assignments, and a set of PPT slides handouts of the lectures.

The book is organized as follows. Chapter 1 provides an overview of database systems. It highlights the key difference between data, information, and knowledge, as well as the key concept of data redundancy. Chapter 2 introduces the important data structure techniques that are commonly used in databases. Chapter 3 describes the entity-relationship model which is a key element of the foundation of database design. Chapter 4 discusses the relational database model. It ties the entity-relationship model with the relational database model. Chapter 5 provides details of normalization process which is another key element of the foundation of database design. Chapter 6 explains SQL, a standard database processing language. Chapter 7 discusses the concept and major techniques of physical database design which is critical for large-scale databases. Chapter 8 discusses the major database administration functions. Chapter 9 discusses the key concept of distributed databases in the Internet environment. Chapter 10 discusses the application of relational databases for business intelligence in the data warehouse context. The book also contains a comprehensive set of student study guides. Technical Guide A is a tutorial of database implementation and application through the use of Microsoft Access. Technical Guide B is a tutorial of data normalization process and ER diagram construction based on data samples. Technical Guide C is a tutorial of OLAP using Microsoft Office. Answers to selected exercise questions, review sheets for preparation for exams, and a set of PPT slides handouts are attached at the end of the book.

Nowadays, students and young managers are the digital generation. They expect to

learn more about information technologies to meet the challenge of the “Big Data Problem” in business. The vast amount of data creates an increasing degree of needs as well as pressure for managers to understand and to use information for their business. We believe that the basic foundation of database technology and data resources management is definitely needed for business students to become successful business professionals.

In summary, this textbook is written for university students in all majors who study database technology for data resource management in business.

Shouhong Wang, PhD
University of Massachusetts Dartmouth

Hai Wang, PhD
Saint Mary’s University

CHAPTER 1. INTRODUCTION

1.1. Database Technology

Database technology includes theory and application methodologies for building information systems that manage, access, and process data. Database technology is critically important for business in many aspects.

- Database technology supports the database development for information systems. The database design is parallel with the business process analysis and design for the information system development.
- Database technology empowers organizations to manage data resources to meet the challenges of the “**Big Data Problem**”. Big data are a collection of data sets that are large and complex. The challenges include capture, preservation, storage, search, sharing, transfer, analysis, and visualization. The major objective of big data analysis is to derive information from analyses of separate but related large sets of data to determine global trends. Generally, big data are characterized by three dimensions: volume (amount of data), velocity (speed of data in and out), and variety (range of data types and sources).
- Database technology enables the society and business organizations to develop organizational knowledge and to sustain organizational learning.

1.2. Data Are Resource of the Organization

Data are valuable resource of the organization. Data can be used for the organization in day-to-day operations as well as in developing competitive advantages. There are many types of data in business. For the time being, no commonly accepted taxonomy of data is available, but the following types of data can be readily observed in business organizations.

- Master data (e.g., customers)
- Transaction data (e.g., sales)
- Historical data (e.g., credit history)
- Secondary data (e.g., industrial publication)
- Subjective data (e.g., end-user survey)

Master data characterize the properties of stakeholders of the organization. They are resident in the system permanently, and are usually used for the daily activities. Transaction data record the transactions of business, and are usually temporary. They might be kept in the system for a certain period for specific purposes such as auditing. Historical data are chronicle records of useful facts. Secondary data are facts obtained from outside sources. Subjective data are human opinions.

One of the critical issues of data resource management is **data quality**. Generally, data quality is measured by many attributes of data, including accuracy, objectivity, believability, reputation, access, security, relevancy, value-added, timeliness, completeness, amount of data, interpretability, ease of understanding, concise representation, and consistent representation. The ultimate objective of databases is to provide quality data for the organization in these data quality attributes.

1.3. Data, Information, Knowledge

Although the distinction between **data**, **information**, and **knowledge** can never be unambiguous, we must differentiate the three for this database course. For our purpose, data are raw facts; information is a product of processed data in a given context; and knowledge is human interpretation of the real world. While tacit knowledge is in the human's mind, explicit knowledge can be represented and stored by computers in a form of information. For instance, sales figures are data which record the facts of sales. "*The total of sales*" is information because it is not a raw fact, but is a processed result of sales of all sales teams over a certain period. A proposition such as "*the low total of sales is a result of the slow economy*" is knowledge, because it is not a raw fact, nor an immediate product of processed data, but is a human perception and judgment. Note that data, information, and knowledge are not interchangeable terms in this book. In principle, a database stores data (not information, nor explicit knowledge), and is able to process the stored data to generate information for the user to develop knowledge. Certainly, a computer can always store information and explicit knowledge, but not in the normal form of database. In fact, "information repository" and "knowledge base" are ambiguous terms for less structured digitalized resources.

1.4. Common Mistakes in Data Resource Management

Data is a valuable resource of the organization. However, many people do not know much about the database technology. They use non-database tools, such as Excel spreadsheet or Word document, to store and manipulate business data, or use poorly designed databases for business processes. Eventually, the data are redundant, inconsistent, inaccurate, and corrupted. The corrupted data storage is actually a garbage can. The common mistakes in data resources management are discussed as follows.

(1) No control of redundant data

People often keep redundant data for convenience. Redundant data could make the data set inconsistent. We use an example to explain why redundant data are harmful. Suppose the registrar's office has two separate files that store student data: one is the registered student roster, and the other is student grade roster, as shown in Figure 1.1. You can find in Figure 1.1 that the fact "*the student holding the student number 1432 is Ann Riley*" is stored in both spreadsheets in multiple places. Such a duplication occurrence that repeats the same fact stored in the data system is called **data redundancy**. Redundant data often make data access convenient, but can be harmful. For example, if Ann Riley changes her name, then all her names stored in the system must be changed altogether. However, practically, making changes to all redundant data is difficult if the data set is huge. As a result of data redundancy, the entire data set can eventually be corrupted.

(2) Violation of data integrity

Data integrity means consistency among the stored data. We use the example in Figure 1.1 to explain the concept of data integrity and how data integrity can be violated if there is no systematic means to maintain data integrity. In Figure 1.1, you can find in the grade roster that Chris Smith received a grade in MKT211; however, you can't find Chris Smith in the student roster. In this example, the two rosters are not consistent. Suppose we have a data integrity control procedure to enforce the rules, say, "no student can receive a grade unless she/he has registered", then such a violation of data integrity can never happen.

The figure consists of two screenshots of Microsoft Excel spreadsheets. The top screenshot shows a spreadsheet titled 'Student Registration Roster' with columns A, B, C, D, and E. The data is as follows:

	A	B	C	D	E
1	Student Registration Roster				
2	Student ID	Student Name	Student Address	Student Year	
3	1334	John Kerry	38 Union Ave.	2017	
4	1432	Ann Riley	17 Wood Lane	2016	
5	2874	Doug Dale	6 Lake Rd.	2018	
6					
7					

The bottom screenshot shows a spreadsheet titled 'Student Grade Roster' with columns A, B, C, and D. The data is as follows:

	A	B	C	D
1	Student Grade Roster			
2	Student ID	Student Name	Course ID	Grade
3	1234	Chris Smith	MKT211	B
4	1432	Ann Riley	ACT201	A
5	1432	Ann Riley	MGT311	A
6	2874	Doug Dale	MKT211	C
7				

Figure 1.1. An Example of Mistakes in Data Resource Management

(3) Relying on human memory to store and to search needed data

The third common mistake in data resource management is the over-use of human memory for data search. A human can remember what data are stored and where the data are stored, but may forget what data have been stored and where these data are stored at any time. If a piece of data is stored in an un-remembered place, it has actually been lost. As a result of relying on human memory to store and to search needed data, the entire data set eventually becomes disorganized.

To avoid the above common flaws in data resource management, database technology must be applied.

1.5. Control Data Redundancy

One of the major objectives of data resource management is to control data redundancy. Data redundancy not only waste resources for data entry and updating, but also causes data inconsistency and thus corrupts the database. For instance, suppose a customer holding a certain customer number has filed her data in five store branches of a retail company, and the customer changes her address. The system must remember to change all addresses in the five branches. If only one address is changed and others remain unchanged, then her addresses are no longer consistent in the data system of the retail company. We will further discuss other problems caused by data redundancy later in this book.

Data redundancy occurs when the same **fact** is stored in more than one place. Note that this definition of data redundancy should not be read like “the same number (or text) is stored in more than one place.” Let us examine the examples in Figure 1.2 to understand more about data redundancy. There are two cases in Figure 1.2, and both show the storages of the customers’ data and purchase data. In case (a), the customer’s data repeat in several places. The **fact** that “*the customer holding the customer number 123456 is Smith who lives in Westport Rd. with phone number 508999*” is stored twice. Hence, case (a) is a case of data redundancy. In case (b), the **fact** that “*the customer holding the customer number 123456 is Smith*” is stored in the CUSTOMER table only once. Thus, it does not have the data redundancy problem occurred in case (a). Take a look at the PURCHASE table in case (b). The same customer number 123456 is stored in the two places. Does data redundancy occur here? The answer is no, because the same customer number 123456 represents the two different **facts** in the two places; that is, the customer 123456 had two purchases. Take a look at both tables in case (b). The same customer number 123456 is stored in both tables. Does data redundancy occur here? The answer is no. In the CUSTOMER table, this number represents the **fact** that “*the customer holding the customer number 123456 is Smith.*” However, in the PURCHASE table, the customer number 123456 represents different **facts** of purchases made by this customer.

CUSTOMER- PURCHASE

CustomerNumber	CustomerName	CustomerAddress	CustomerPhone	Purchase	PurchaseTime
123456	Smith	Westport Rd.	508999	\$200	10/20/2014 09:45am
234562	Green	Eastport Ave.	509343	\$300	10/23/2014 01:32pm
123456	Smith	Westport Rd.	508999	\$150	11/08/2014 05:04pm

(a) Data redundancy occurs

CUSTOMER

CustomerNumber	CustomerName	Customer Address	CustomerPhone
123456	Smith	Westport Rd.	508999
234562	Green	Eastport Ave.	509343

PURCHASE

CustomerNumber	Purchase	PurchaseTime
123456	\$200	10/20/2014 09:45am
234562	\$300	10/23/2014 01:32pm
123456	\$150	11/08/2014 05:04pm

(b) There is no data redundancy if the data are stored in this form

Figure 1.2. Control Data Redundancy

This example shows that a system can avoid data redundancy, as long as the data are stored in an appropriate way. On the other hand, this example also shows the potential difficulty of **data integration** when the data has no redundancy. For instance, more likely, the user of the data prefers the integrated data as shown in case (a) of Figure 1.2 to find all associated facts in just one table. In case (b), the user has to search the two tables and merge them together every time she wants to obtain the integrated data. If the organization has a huge number of tables, it is impossible for human to perform such tedious jobs. This example explains why computerized database systems are needed to achieve no-redundancy as well data integration for data

resource management.

1.6. Database and Database System

Organizations use database systems to manage data efficiently and effectively in the following aspects.

- (1) Controlled data redundancy - All data files are integrated into a single logical structure of database. In principle, any fact is recorded once.
- (2) Data consistency - As a result of data redundancy control, data in database are consistent.
- (3) Data sharing - All authorized users in the organization share the data in the database.
- (4) Facilitate application development - A database system provides powerful abilities to access the data in the database.
- (5) Wide-ranging data management functions - Database management systems provide integrity control, backup and recovery, and security and privacy control functions.

Key terms of database system are explained as follows:

- **Database (DB)** - A database is a collection of data which is accessed by more than one person and/or which is used for more than one purpose.
- **Database management system (DBMS)** - A database management system is a collection of procedures, languages and programs which can be used to facilitate the design, implementation, operation, and maintenance of the database system.
- **Data model** - The data model of a database is the logical structure of the data items and their relationships.
- **Database administration procedures** - The formal policies and procedures for the data resource management in the organization.
- **Database system** - A database system consists of the database, the DBMS, the computer hardware, **database administrator (DBA)**, and the database administration procedures for the database.

The general architecture of a database system is illustrated in Figure 1.3.

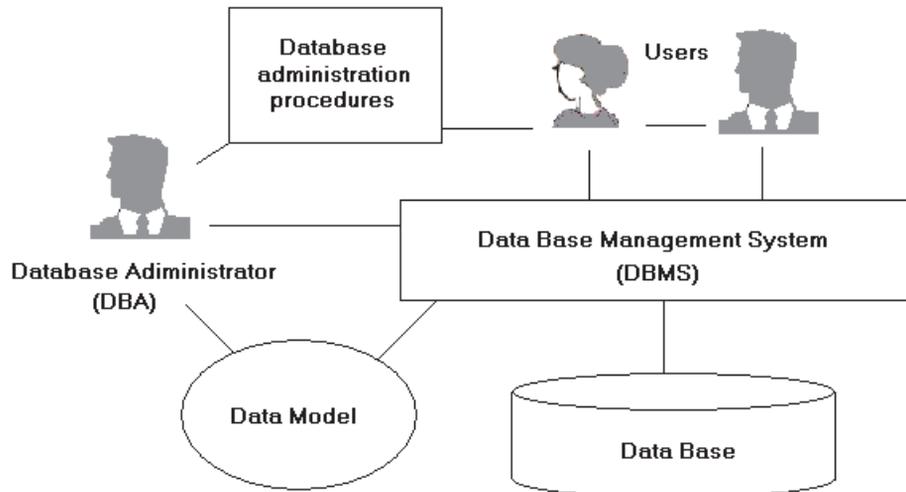


Figure 1.3. Database System

1.7. Database Management Systems

To create and manage a database, a **database management system (DBMS)** must be used. A DBMS is a software system which can be used to facilitate the design, implementation, operation, and maintenance of database systems. The key concepts of DBMS and common functions of DBMS are discussed as follows. Note that an individual DBMS has its own algorithms and procedures to support those functions at its own sophistication level.

1.7.1. Support database construction and data access

A DBMS provides the **interface** between the user and the computer to allow the user to create a database, to input data, to update the database, and to access the data.

1.7.2. Control data redundancy

In principle, there should be no data redundancy in a database. However, in a large

database, no redundancy will result in poor performance (i.e., slow response). As explained later, a large database might have controlled data redundancy. Such data redundancy might be temporary and has a limited scale. One of the DBMS functions is to ensure that the controlled data redundancy does not compromise the data **accuracy** and **integrity**.

Data integrity means **consistency**; that is, there are no conflicting facts in the database. **Data accuracy** and **verifiability** are exchangeable terms of data integrity.

A redundant data set often causes **data modification anomaly** and violates data integrity. Chapter 5 will further explain data modification anomaly. The DBMS eliminates the data redundancy and avoids the data modification anomaly.

1.7.3. Provide data integration

Data **integration** refers to the ability to tie together pieces of related data upon the request from the user. For instance, the user might like to have all data about “*the courses John Smith has taken, the grades he has received from these courses, and the instructors of those courses*” in a single table instead of searching several tables. Usually, integrated data contain data redundancy, and are not supposed to be stored in the database permanently. One of the DBMS functions is to support data integration by assembling the available data in the database for the user while controlling the data redundancy.

1.7.4. Maintain data independency

Data are processed by computer programs. On the other hand, the format of data and the way in which the data are stored on the media (disks) are highly dependent on the individual database. One of the DBMS functions is to ensure the data **independency** so that any changes in the data storage within the database do not require changes in the application computer programs.

1.7.5. Monitor and improve data retrieval performance

As will be explained later, performance in a large database is crucial for the success of the information system, and must be taken into account for the **physical database design**. The DBMS monitors the data retrieval performance and provides information for the DBA for the database physical design. A good DBMS has the

capacity to support database physical design, or even automatically take actions to improve the data retrieval performance. We will further discuss physical database design in Chapter 7.

1.7.6. Control data security

Data security is an important issue of database. The DBMS acts as a security guard for the database. The DBMS applies the **authentication** process to verify that only registered legitimate users are allowed to access the database. It applies the **authorization** process to determine whether the user is allowed to read, to change, or to delete a specific piece of data.

1.7.7. Enforce business rules and maintain data integrity

Many **business rules** related to the data can be built-in the database through defining cardinality, modality, data access security, etc. For instance, “*one professor can have only one office*” is a business rule. If the DBA builds this business rule in the database through the definition of the cardinality, the DBMS will not allow the database to have a fact that is contradictory to this business rule, and maintains the data integrity.

1.7.8. Manage concurrency control

When multiple users access and update the same piece of data concurrently, the data could become incorrect. This is caused by the time delays between the data access and the data updating among the multiple users, which is known as the **concurrency** problem. A sophisticated DBMS has advanced techniques and procedures, called concurrency control protocols, to manage concurrency control to ensure the correct and prompt data access and data updating by multiple users in the data sharing environment.

1.7.9. Perform backup and recovery

Databases are vulnerable and could be damaged by fire, flood, earthquake, and other human-made or natural disasters. A good DBMS is able to perform **backup** from time to time automatically, and to provide instructions for **recovery** when the database is damaged. We will further explain backup and recovery later in this book.

1.7.10. Maintain data dictionary

A database has a **data dictionary**, as further explained later in Chapter 8. A data dictionary contains data about the database. The DBMS maintains the data dictionary for the data resource management.

1.7.11. Facilitate database restructuring

The business environment keeps changing, so does the database. A DBMS can facilitate offloading, redefinition, reloading, conversion, etc. to accommodate the changes.

1.8. Commonly Used DBMS for Relational Database

There are many commonly used DBMS for relational database in business.

(1) ORACLE is Oracle Corporation's product. It first appeared around 1979, and was one of the first relational DMBS in the IT industry. It has been widely used in business since then.

(2) IBM DB2 (or DB2) is IBM's product. It was also a pioneer of relational DBMS in the early 1980s. DB2 is the first commercialized database product that uses SQL (Structured Query Language) which was also developed by IBM.

(3) MySQL was released in 1995. Later, MySQL becomes open-source software under the GNU General Public License (GPL). MySQL is a popular choice of database for Web applications, because it is closely tied to the popularity of PHP, an open-source server-side programming language.

(4) Microsoft SQL Server was Microsoft's entry to the enterprise-level database market, competing against ORACLE and IBM DB2 in about 1989. It is a widely used DBMS in many enterprises.