

Java 语言高级教程

孙一林 主编

清华大学出版社

(京) 新登字 158 号

内 容 简 介

Java 是一种具有平台独立性的编程语言，它在源程序级上保证其基本数据类型与平台无关。Java 编程语言的原则是“一次编写，随处运行”，它最适合于分布式的网络编程。Java 编程从根本上改变了网络应用程序的开发和使用方式，并成功地应用于网络电子商务、物流配送等“B to B”、“B to C”的系统中。本书是一本基于网络（Internet、Web 等）应用系统而指导 Java 编程的教材。

本书主要阐述了 Java 语言在服务器端的各种方式的应用。主要内容包括：Java 数据库应用系统的建立、服务器和客户端数据库操作系统的安装和配置、使用 Java 语言实现网络三层数据库的应用、服务器端数据库应用实例以及 JavaBean 组件技术等。本书从基础原理到使用方法，介绍了构建 Java 数据库系统和使用组件技术完成系统工程的建设。配书光盘中列举了大量实用的实例，每个实例都可以作为一个组件而应用于项目工程中，所有的实例都是在 JBuilder 开发环境中编译通过的。

本书面向具有一定 HTML 和 Java 语言基础的读者，可以作为大学本科 Java 语言基础课程的后续高级课程的教材，也可用于研究生教学中，同时还可作为使用 Java 构建电子商务等 Web 应用技术的参考书。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

书 名：Java 语言高级教程

作 者：孙一林 主编

出 版 者：清华大学出版社（北京清华大学学研大厦，邮编 100084）

<http://www.tup.tsinghua.edu.cn>

责任编辑：钟志芳

印 刷 者：

发 行 者：新华书店总店北京发行所

开 本：787×1092 1/16 印张：25.75 字数：589 千字

版 次：2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

书 号：ISBN 7-900643-62-1

印 数：0001 ~ 5000

定 价：33.00 元（附光盘）

前 言

近几年来,作者和参与《Java 语言高级教程》编写的同仁一直从事用 Java 语言开发分布式网络应用系统的项目,以及 Java 语言的教学工作。在项目的研发和教学过程中,对 Java 语言的应用深有体会。作者编写本教材的目的是对以往使用 Java 语言的体会作系统全面、细致的总结。

Internet 的普及使得电子商务的实现变成了可能,构建跨平台的电子商务系统已经越来越受到业内人士的重视。本书就是以普及 Java 语言在网络中的应用为宗旨而编写的。在编写的过程中,力求做到由浅入深、深入浅出,所有实例都具有实用性(部分实例是从实际应用项目中提炼出来的),能引导读者更好地掌握书中的内容,使读者学有所用。

本书由北京师范大学计算机系孙一林副研究员编写。在编写该教材的过程中,得到了中国农业大学计算机系系主任彭波副教授的协作,彭教授参与了有关数据库基础知识部分的编写;得到了中国科学院软件所李小满硕士的大力支持,为作者提供了《北京交通查询系统》等实例;还得到了北京师范大学信息科学学院王建明副教授和北京师范大学计算中心张庆利工程师的帮助,他们调试了本书中大部分有关数据库方面的实例;得到了北京师范大学数学系吴慧博士和曾跃进工程师的帮助,他们为调试书中的实例构建了 Java 服务器实验系统,并调试了部分实例;得到了东方易维网络技术公司梁言兵硕士的帮助,他为作者提供了有关 JavaBean 组件的基础知识并参与了组件实例的调试。此外,作者的研究生吕小晴、王茜同学也参与了部分实例的调试工作,在此一并向他们表示感谢。

在编写本书的过程中,疏漏和不当之处,敬请广大读者指正,如果读者在阅读时发现了问题,请发送电子邮件到:sunyilin@elec.bnu.edu.cn。

作 者

2002 年 5 月于北京



第 1 篇 Java 数据库程序设计

第 1 章 数据库技术概述.....	1
1.1 数据库技术特点.....	1
1.2 数据模型.....	1
1.3 数据库系统应用结构.....	7
1.4 数据库管理系统.....	10
1.5 数据库系统应用设计.....	11
1.6 安装和运行 InterBase 数据库系统.....	15
1.7 习题.....	22
第 2 章 SQL 语言.....	23
2.1 SQL 数据定义语句.....	24
2.2 SQL 数据查询、更新语句.....	27
2.3 SQL 数据控制语句.....	29
2.4 SQL 语言应用举例.....	32
2.5 习题.....	36
第 3 章 JDBC 编程接口.....	38
3.1 ODBC 概述.....	38
3.2 JDBC 概述.....	42
3.3 InterClient 纯 Java JDBC 驱动程序.....	48
3.4 习题.....	51
第 4 章 JDBC 类和接口.....	53
4.1 JDBC API 类和接口.....	53
4.2 JDBC API 常用类.....	54
4.3 JDBC API 常用接口.....	63
4.4 习题.....	80
第 5 章 建立 Java 数据库应用模型.....	81
5.1 创建数据源.....	81

5.2	操纵数据库.....	83
5.3	建立网络数据库应用模型.....	95
5.4	测试数据库.....	103
5.5	习题.....	104
第 6 章	JBuilder 数据库组件应用.....	107
6.1	JBuilder 数据库浏览工具.....	107
6.2	JBuilder 组件操纵数据库.....	110
6.3	JBuilder 数据模块.....	120
6.4	JDBC 驱动程序监视器.....	128
6.5	习题.....	131

第 2 篇 Java Web 服务器

第 7 章	Servlet 服务器小程序.....	132
7.1	Servlet 小应用程序结构.....	132
7.2	Servlet 小程序服务器端运行环境.....	136
7.3	创建和运行 Servlet 小应用程序.....	143
7.4	Servlet 连接数据库.....	156
7.5	习题.....	167
第 8 章	JSP 动态网页技术.....	169
8.1	JSP 网页基础知识.....	169
8.2	JSP 服务器端运行环境.....	179
8.3	创建和运行 JSP.....	184
8.4	JSP 连接数据库.....	189
8.5	习题.....	200
第 9 章	Internet 数据库应用实例.....	203
9.1	Applet 公交线路查询系统.....	203
9.2	Servlet 留言板小程序.....	215
9.3	网络聊天室.....	221
9.4	习题.....	240

第 3 篇 JavaBean 组件设计

第 10 章	JavaBean 组件.....	241
10.1	基于组件技术的软件开发.....	241
10.2	JavaBean 基本概念.....	242

10.3	JavaBean 开发和运行环境.....	245
10.4	创建 JavaBean 组件.....	246
10.5	习题.....	260
第 11 章	JavaBean 测试工具 BDK.....	261
11.1	安装和启动 BDK.....	261
11.2	JavaBean 组件测试.....	263
11.3	BDK 构造应用程序.....	274
11.4	习题.....	275
第 12 章	JavaBean API	276
12.1	JavaBean API 类和接口	276
12.2	JavaBean API 常用类	277
12.3	JavaBean API 常用接口	285
12.4	习题.....	290
第 13 章	JavaBean 属性.....	291
13.1	Simple 属性	291
13.2	Index 属性.....	293
13.3	Bound 属性	294
13.4	Constrained 属性.....	302
13.5	JavaBean 组件属性的其他特性.....	309
13.6	习题.....	323
第 14 章	JavaBean 事件.....	325
14.1	JavaBean 组件事件模型.....	325
14.2	JavaBean 组件事件对象.....	326
14.3	JavaBean 组件事件监听器.....	327
14.4	JavaBean 组件事件源与目标对象.....	333
14.5	JavaBean 组件事件适配器.....	339
14.6	JavaBean 组件应用举例.....	345
14.7	习题.....	356
第 15 章	Enterprise JavaBeans 入门.....	357
15.1	编写 EJB 组件	357
15.2	会话 Bean.....	362
15.3	实体 Bean.....	368
15.4	建立 EJB 组件开发环境	372
15.5	可视化创建、测试、运行 EJB 组件	375

15.6 习题.....	386
附录 A J2SDK 文档使用.....	387
A1 javac.....	388
A2 java.....	388
A3 appletviewer.....	389
A4 javadoc.....	391
A5 jar.....	393
A6 jdb.....	394
A7 javap.....	395
A8 extcheck.....	396
A9 rmic.....	396
A10 rmid.....	397
A11 rmiregistry.....	398
A12 jarsigner.....	398
A13 Keytool.....	399
附录 B 配书光盘内容介绍.....	400

第 1 篇 Java 数据库程序设计

第 1 章 数据库技术概述

数据库技术是研究数据库的结构、存储、设计、管理和使用的软件学科，它是计算机科学的重要分支。数据库系统是采用数据库技术的计算机系统，它是信息技术和信息产业的基础。目前，信息资源已经成为一种重要的财富，从小型事务处理系统到大型信息共享系统，越来越多的新的应用领域采用数据库技术对信息资源进行存储、处理以及利用。正是由于数据库技术的不断发展才促进了社会信息化的进程。

随着 Internet 网络的发展和应用的普及，利用数据库技术创建信息丰富、与客户端交互性强的 Internet 站点成为必然。

1.1 数据库技术特点

数据库是通用化的综合性的数据集合。数据库技术是以共享数据库为中心来研究和开发数据库应用系统的一项技术。以数据库为核心开发数据库系统的原则有利于数据的集中管理以及应用程序的研制和维护，它可以提高数据的利用率、可靠性以及相容性等。通用的数据库技术应具有以下特点：

1. 面向复杂的数据结构。
2. 数据冗余度小，易于扩充。
3. 具有较高的数据与程序的独立性。
4. 统一的数据控制功能，其中包括数据的安全性、完整性以及并发控制。
5. 以单一数据项为数据的最小存取单位。

1.2 数据模型

现实世界中的数据在数据库系统中是通过数据模型来进行抽象表示和处理的。所有的数据库系统均是基于某种数据模型建立的。数据模型是数据库技术的核心，其模型的优劣直接影响数据库的性能。

1.2.1 数据库技术基本术语及概念

• 数据和信息：数据是对客观事物属性的描述与记载，是一些物理符号。信息则是人们对数据进行加工整理后，为某一特定目的所提供的决策依据。信息源于数据。

• 数据处理：数据处理是对各种类型的数据进行收集、存储、分类、计算、加工、检索和传输的过程。

• 数据库 (DataBase, DB)：数据库是长期存储在计算机内，有组织、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和存储，具有较小的冗余度，较高的数据独立性和易扩展性，并且可以为各种使用者所共享。数据库是独立于数据库应用程序的。

• 数据库管理系统 (DataBase Management System, DBMS)：数据库管理系统是位于数据库使用者与数据库操作系统之间的一层数据管理软件，它是由系统运行控制程序、语言翻译程序和一组公用程序所组成。

• 数据库系统 (DataBase System, DBS)：数据库系统通常是指带有数据库的计算机系统，包括数据库相关的硬件和软件。

• 实体 (Entity)：实体是用来描述客观事物的，它可以指事物本身，也可以指事物之间的联系，它是采集数据的对象。例如，学生群体为一个实体。

• 属性 (Attribute)：属性是对实体有确切意义的特征的抽象描述，一个实体可以由若干个属性来描述。例如，学生的属性有学号、姓名、年龄、性别等。

• 主键 (Key)：主键是实体属性集中的一个惟一标识。例如，定义学生实体属性集中的学号为主键，所以学号是在学生实体中惟一代表每个学生的属性。

• 实体型 (Entity Type)：实体型是用实体名和属性名的组合来抽象和描述同一类型的实体。例如，学生 (实体名) 和学生的学号、姓名、年龄、性别和班级等 (属性名) 是一个实体型。

• 实体集 (Entity Set)：实体集是同一类型实体的集合。例如，一个学校的所有学生组成一个实体集。

• 联系 (Relationship)：联系是指实体之间或者属性之间的内在关系。在数据库系统中，实体之间 (或者属性之间) 的联系被划分为三种类型。

1. 一对一联系 (1:1)：相对实体集A中的每一个实体，实体集B中只有一个实体与之联系，反之亦然，则实体集A与实体集B具有一对一联系。记作1:1。

2. 一对多联系 (1:n)：相对实体集A中的每一个实体，实体集B中有n个实体 ($n \geq 1$) 与之联系，反之，对于实体集B中的每一个实体，实体集A中只有一个实体与之联系，则实体集A与实体集B具有一对多联系。记作1:n。

3. 多对多联系 (m:n)：相对实体集A中的每一个实体，实体集B中有n个实体 ($n \geq 1$) 与之联系。反之，相对实体集B中的每一个实体，实体集A中也有m个实体 ($m \geq 1$) 与之联系，则实体集A与实体集B具有多对多联系。记作m:n。

【示例 1.1】例举在学校中出现的 1:1、1:n、m:n 实例。

一个学校有一个校长，每个校长只在一个学校任职，所以，学校和校长之间是一对一联系；一个系里有很多学生，每个学生只在一个系里学习，所以，系与学生之间是一对多联系；一门课程有很多学生选修，而每个学生也要选多门课程，所以，课程与学生之间是多对多联系。该示例实体之间的联系如图 1.1 所示。

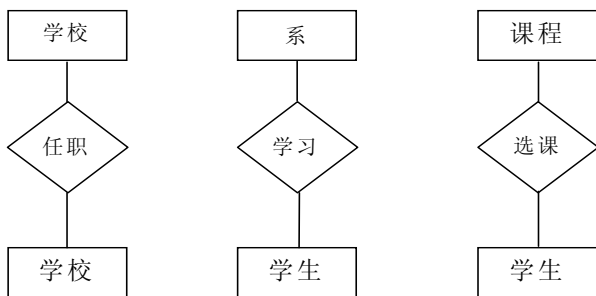


图 1.1 实体之间的联系

1.2.2 数据模型基本概念

数据模型是严格定义的一组概念的集合。这些概念精确地描述了系统的静态特性、动态特性和完整性约束条件。数据模型通常由数据结构、数据操作和完整性约束条件三部分组成。

1. 数据结构

数据结构是数据库技术所研究的对象类型的集合。这些对象也是数据库的组成部分，一般可分为两类：一类是与数据类型、内容、性质有关的对象；另一类是与数据之间联系有关的对象。

在数据库系统中通常按照数据结构的类型来命名数据模型。例如，层次结构、网状结构和关系结构的模型分别命名为层次模型、网状模型和关系模型。

2. 数据操作

数据操作是指对数据库中各种数据对象实例（即值）所允许执行的操作的集合，包括操作和操作规则。数据库操作主要划分为两大类：检索和更新（插入、删除和修改）。数据模型需要定义这些操作的确切含义、操作符号、操作规则（优先级别）以及实现操作的语言。

如果说数据结构是对数据库系统静态特性的描述，那么数据操作则是对数据库系统动态特性的描述。

3. 完整性约束条件

数据的完整性约束条件是指完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的制约和依存规则。这些规则用来限定基于数据模型的数据库的状态

及状态的变化，以保证数据库中数据的正确、有效和相容。

数据的完整性约束条件分为以下三类：

- 实体完整性规则。例如，规定关系中的主键值不能为空值。
- 参照完整性规则。例如，规定关系中每个非空的外键需要与另一关系中的主键相匹配（键值的属性）。
- 数据库使用者根据数据模型提供的完整性约束条件定义自己的完整性规则。它反映了某一具体应用所涉及的数据应满足的语义要求。

此外，数据模型还需要提供定义完整性约束条件的机制，以反映某一应用所涉及的数据遵守的特定的语义约束条件。

1.2.3 关系模型数据库

在数据库应用系统中，数据的模型主要有三种形式：层次模型、网状模型和关系模型。其中，层次模型和网状模型称为非关系模型，而关系模型构成关系数据库。

层次模型是用树型拓扑结构表示实体间联系的模型。树型结构由结点和连线组成，结点表示实体集，连线表示实体之间的联系。层次数据库采用层次模型作为数据的组织方式，它自然、直观、易于理解。

网状模型通过网状拓扑结构描述实体间的联系，网状数据库采用的是网状模型作为数据的组织方式。任意一个连通的基本层次联系的集合是一个网状模型。

关系模型数据库是目前应用最为广泛的一种数据模型。关系模型是通过满足一定条件的二维表来表示的数据及其数据之间的联系的一种模型。

1. 关系模型中的基本概念

- 表：表是由若干行和列（ m 行和 n 列）组成的数组，即一个二维表格。
- 关系：一个关系对应一个二维表，关系即表与表之间的联系。数据库可以由多个表组成，表与表之间可以以不同的方式相互关联。
 - 字段：数据库表中的每一列称为一个“字段”（属性），每个表都有若干个字段列，每个字段描述了表的一个属性值，每个字段列有唯一的字段名称。字段可以包含各种字符、数字甚至图形。
 - 域：域是字段中的数据取值范围。
 - 记录：存放在表中的一行称为一个“记录”（元组）。每个表由多个同类记录组成。同一个数据表中任意两个记录都不能完全相同。
 - 键（主键、外键）：键是在表中被指定的某个（或多个）字段，键在表中可以是唯一的，也可以是非唯一的。在数据库系统中键有主键和外键之分，主键（关键字）是一个表（实体属性集）中的一个唯一标识，用来唯一表示一个表的每一行，即在一个表中的某个字段组的值能够唯一地表示一个记录。外键是在一个表中参照另外一个表中的一个字段定义的键值，它使得两个表之间可以发生联系。如果一个表中的字段被定义为一种键，则该字段的数值应该是非空值（NOT NULL）。

- 分量：分量是记录中一个字段的值。它是表中不可再分的最小数据单位。
- 关系模式：关系模式是对关系的描述。关系模式包括关系名和组成该关系的属性名。通常记为：关系名（属性名 1，属性名 2，…，属性名 n ）。

2. 关系模型的性质

- 关系中每一个属性（字段）值都是不可分解的数据项，并取自同一个域。
- 每个记录中同一区域中的属性具有相同的数据类型。
- 关系中不允许出现相同的记录（没有重复记录）。
- 表中行/列的顺序不固定，行/列的次序可以任意交换。

3. 关系模型的特征

- 结构单一化。在关系模型中，基本数据结构被限制为二维表格。
- 数据结构简单明了，便于使用者了解和维护。
- 数据库使用简便，便于数据库应用程序的开发。
- 数据独立性好，有较好的一致性和良好的保密性，并体现出高效率的数据处理。
- 数据库的数据存取不必依赖于索引，可以自定义优化算法。
- 可以配备多种高级语言接口操纵数据库。

4. 关系数据库

关系数据库是由一系列相互关联的表所组成，即将数据表示为表的集合，它是通过建立简单表之间的关联来定义结构的一种数据库，在每一个表中存有一类与应用系统相关的数据。例如，一个在校学生数据库中，可能有关于学生情况、联络方式、选课情况等方面的表。在数据库中，每一个这样的实体将被作为一系列的字符串、整数及其他原始数据类型存储起来。在数据库中，表的定义将描述每一种与实体相关的信息如何在一个表的字段中存储，例如，学生基本情况表中“姓名”字段等。每一张表应有一个或几个字段值作为标识（键），确保每条记录的惟一性。这些标识或“键”可以用来连接存在于不同表中的信息。例如，在学生基本情况表中，为每一个学生指定惟一的“学号”的键值（主键），并在学生联络方式表中的相应字段中使用同一个键值，即可以通过对两个表中的“学号”字段值的匹配，使每一个学生和他的联系方式关联起来。

【示例 1.2】创建一个关系数据库，数据库中包含学校学生基本情况表、学生联络方式表和学校课程计划表。首先用关系模式的表达式描述三个表并在每个表中设置主键和外键，其次，在三个表之间建立联系。

学生基本情况表关系模式（主键为学号）：

学生基本情况表（学号、姓名、年龄、性别、系名）（见表 1.1）

学生联络方式表关系模式（主键为编号、外键为学号）：

学生联络方式表（编号、E-mail 地址、电话、学号）（见表 1.2）

课程计划表关系模式（主键为课程代号）：

课程计划表（课程代号、课程名、学分、学时、选修课程代号）（见表 1.3）

表 1.1 学生基本情况表

学号(主键)	姓名	年龄	性别	系名
200166001	赵明	20	女	计算机系
200166002	钱晓	21	男	物理系
200166003	孙晨	19	男	计算机系
...

表 1.1 中的关系名为“学生基本情况表”，表中的每一行是一个学生的记录，表中的学号、姓名、年龄、性别、系名均为字段（属性），其中学号唯一地标识一个记录，即“主键”；对于每个字段都有其取值范围（域），例如，年龄的域为一个整型量。

表 1.2 反映了学生的联络方式。在表 1.2 中编号被定义为主键，学号被定义为外键。在一个表中可以定义多个键，但是一定需要有一个主键具备唯一性，其他键可以具有重复性。表 1.2 中编号主键被定义为一个具有唯一标识的属性，在表中被定义为键的字段是不能为空值的，即在一个记录中具有主键或键性质的字段一定要有分量。

表 1.1 学号字段可以同表 1.2 学号字段建立一对多的联系，即一个学生可以有多个 E-mail 地址或电话号码，而一个 E-mail 地址或电话号码只会同个确定的学生相对应。由于表 1.1 学号和表 1.2 学号建立了某种方式的关联，当一个学生从表 1.1 中被删除后，则与之对应的表 1.2 该学生的相关记录通过关系数据库的特定操纵也可以同时被删除，因为学生已从数据库中删除。所以，在数据库中与该学生相关的一些信息已经没有实际意义了。

表 1.2 学生联络方式表

编号(主键)	电话	E-mail 地址	学号(外键)
1	010-62000001	zhaoming@263.net	200166001
2	010-63000001	qianxiao@elec.bnu.edu.cn	200166002
3	010-63000002	qianxiao@bnu.edu.cn	200166002
4	010-64000001	suncheng@bnu.edu.cn	200166003
...

表 1.3 课程计划表

课程代号(主键)	课程名	学分	学时	选修课程代号
A01	高等数学	6	100	B01
A02	计算机	8	160	B02
A03	英语	10	200	B03
...

表 1.3 表示了各门课程的计划。表 1.3 中课程代号被定义为主键，并可以同表 1.1 建立一对多或多对多联系。即一个学生可以选修多门课程，而一门课程同样可以被多名学生选修。

不同的关系不能并存于同一个表中，但在不同的表中可以有相同的属性（字段），例如，“学生情况表”和“学生成绩表”中的“学号”字段，该字段将两个表关联起来，可构成一对一或一对多联系。

关系数据库采用关系模型作为数据的组织方式。在一个关系数据库中，关系模式是相对稳定的，而关系是不断变化的，因为数据库中的数据是不断更新的。

5. 关系操作

用于关系数据库的关系操作方式的特点是集合操作，操作的对象和结果都是集合。在关系模型中操作的对象是关系，操作的结果是关系。在关系数据库中操作的对象是表，操作的结果也是表。

关系模型的基本操作有：

- 选择（Select）：在关系中选择满足指定条件的记录组成新的关系。
- 投影（Project）：在关系中选择若干字段组成新的关系。
- 连接（Join）：在两个关系中选取字段间满足一定条件的记录组成新的关系。

6. 数据库中关系模型的特点

- 关系模型的概念单一

无论是实体还是实体之间的联系都用关系来表示，关系之间的联系则是通过相容（来自同一个域）的属性来表示。关系模型中数据的逻辑结构在数据库使用者眼里就是表。表概念单一、清晰。

- 关系是规范化的关系

所谓规范化是指关系模型中，每一个关系模式要满足一定的要求或者称为规范条件，规范条件很多，但最基本的要求是每一个分量是一个不可分的数据项，即不允许表中还有表。

- 通过关系生成新表

在关系模型中，数据库使用者对数据的检索操作不过是从原来的表中得到一张新的表。即无论是原始数据还是结果数据都是同一种数据结构（二维表）。由于在关系模型中数据操作是集合操作，所以，操作对象和结果是若干记录的集合。在关系模型中，数据库使用者只需要指出“干什么”或“找什么”，而不必详细说明“怎么干”或“怎么找”，这样，既提高了数据的独立性，又提高了数据库使用者的工作效率。

1.3 数据库系统应用结构

数据库系统是指计算机系统中引进数据库之后所构成的系统。数据库产品例如 Oracle、InterBase、Sybase、Microsoft Access、Microsoft SQL Server 等虽然它们的数据

模型不同，使用的语言不同，存储的结构不同，但是，使用这些数据库产品组成的数据库系统所达到的目的是相同的。目前，数据库系统已经应用于各个领域，在实际应用中，数据库系统是多种多样，例如，单机应用、网络应用等。随着因特网的发展以及因特网应用的普及，各种数据库系统应用于因特网则成为必然。

1.3.1 数据库系统组成

数据库系统由数据库、硬件、软件和数据库应用程序四个主要部分组成，带有数据库的计算机系统的硬、软件层次如图 1.2 所示。

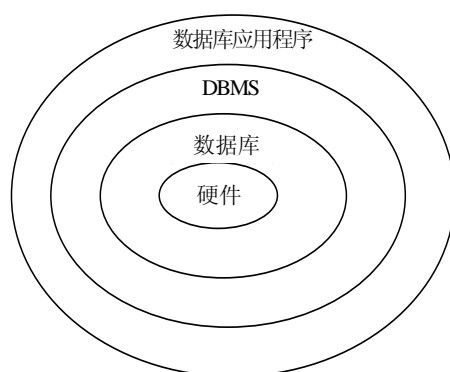


图 1.2 数据库系统层次图

1. 数据库

数据库是具有一定结构的相关数据的集合。系统中所有的数据都存放在数据库中。

2. 硬件

硬件是构成计算机系统的各种物理设备，包括存储数据所需的外部设备。

硬件的配置应能满足整个数据库系统的需要。

3. 软件

物理数据库（实际存储的数据）和数据库系统使用者之间是软件层。该层主要包括支持 DBMS 运行的操作系统以及 DBMS 本身和数据库应用程序。

4. 数据库应用程序

数据库应用程序通过界面（菜单、表格操作、图形显示、报表书写）对数据库进行操作及管理。

1.3.2 三层结构数据库系统

在早期的数据库应用系统中，传统的客户机/服务器（C/S）两层体系结构的数据库系统占据了数据库系统应用中的主导地位。例如，基于 LAN（局域网）数据库系统，其数据库和数据库管理软件驻留在一个专用的服务器机器上，而数据库使用者则通过工

作站上的客户机软件来访问数据库。因为 C/S 两层数据库系统存在着服务器的伸缩性、数据库系统管理、客户机软件代码大小限制和程序复杂性以及无法预测数据库访问数量等问题，所以，数据库系统的应用在逐渐脱离常规的客户机/服务器两层模式，而转向三层体系结构的模式，三层数据库系统摆脱了两层体系结构的限制。并成功地应用于 Internet 网络上。

在 Internet 网络上实现的 Web 应用由于具有统一的用户界面（浏览器）、客户端无须开发任何应用程序已经成为现在数据库系统应用程序开发的一个方向。基于 Internet 网络的 Web 数据库系统应用是以 Browser/Web Server/Database Server（B/W/D 客户机/Web 应用程序服务器/数据库服务器）三层形式出现的，典型的使用 Java 语言开发并应用于 Internet 网络的 B/W/D 三层结构的数据库应用系统结构如图 1.3 所示。

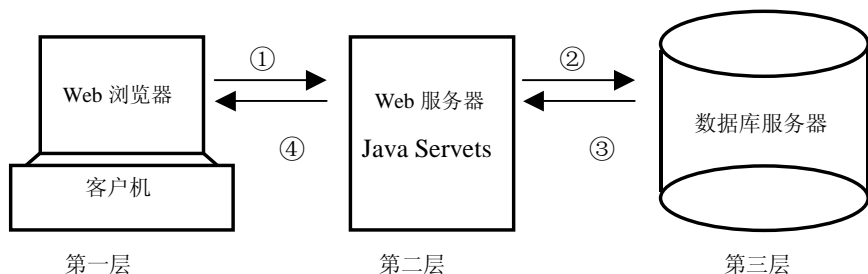


图 1.3 B/W/D 三层结构数据库系统

第一层为客户端（客户层）。使用通用 Internet 网络浏览器，嵌入在 HTML 文档的 Java 语言程序通过 Web 服务器与数据库进行数据交换。

第二层采用运行 Java 语言服务程序的 Web 服务器（中间层）。该层支持应用程序服务器访问数据库，并将获得的数据结果以 HTML 页面形式返给客户层。由于三层数据库体系结构是建立在 Internet 网络 Web 处理模式上，所以，中间层应用程序服务器常被视为 Web 服务器的一种功能扩展。

第三层是后端数据库服务器（数据层）。在该层中驻留了需要保存的数据库数据和数据库操作系统。

在三层结构数据库体系中，数据库应用系统被划分成三个不同的层次。客户端与数据库服务器中的数据交互由四个（①②③④）过程来完成。在该应用模型中，客户通过浏览器调用 Java Applet 程序，Java Applet 程序调用 JDBC API 提出 SQL 请求（过程①），该请求首先传送给提供调用 Java Applet 程序的 Web 服务器，在服务器端通过 JDBC 与指定数据库服务器上的数据库进行连接（过程②），由数据库服务器处理 SQL 语句，然后将结果返回给 Web 服务器（过程③），最后由 Web 服务器将结果发送给客户端（过程④），数据库访问者在浏览器中阅读获得的数据结果。

在三层结构数据库体系应用中，数据库具有较高的物理和逻辑的独立性，从而使数据库获得了良好的性能，通过中间层安全控制，使数据库得到更好的安全保证。另外，