

SQL 语言及其 在关系数据库中的应用

鲍永刚 张英福 王德高 编著



科学出版社

www.sciencep.com

SQL 语言及其在关系 数据库中的应用

鲍永刚 张英福 王德高 编著



科学出版社

北京

内 容 简 介

本书主要介绍 SQL 语言的典型版本以及 SQL 语言在开发工具中的应用方法, 内容全面, 叙述详细。

全书共八章, 内容包括数据库概述、SQL 语言、T-SQL 语言、PL/SQL 语言、关系数据库设计、VB.NET 和 Visual C#数据库连接与 SQL 操作、PowerBuilder 数据库连接与 SQL 操作、Java 数据库连接与 SQL 操作等。本书各章附有大量的应用实例, 实例程序都经过作者实际调试, 均可成功运行, 实用性很强。读者可通过实例, 举一反三, 巩固所学内容。

本书可供数据库应用软件开发人员参考阅读, 也可作为高等院校计算机相关专业教材或教学参考书。

图书在版编目 (CIP) 数据

SQL 语言及其在关系数据库中的应用/鲍永刚, 张英福, 王德高编著.
—北京: 科学出版社, 2007

ISBN 978-7-03-018265-4

I. S… II. ①鲍…②张…③王… III. 关系数据库-数据库管理系统, SQL-程序设计 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2006) 第 148899 号

责任编辑: 岳亚东 刘晓融 / 责任制作: 魏 谨

责任印制: 刘士平 / 封面设计: 李 力

北京东方科龙图文有限公司 制作

<http://www.okbook.com.cn>

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

源海印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2007 年 1 月第 一 版

开本: B5(720×1000)

2007 年 1 月第一次印刷

印张: 18

印数: 1—4 000

字数: 347 000

定 价: 30.00 元

(如有印装质量问题, 我社负责调换〈新欣〉)

前 言

绝大多数软件开发项目都涉及数据库应用。数据库应用开发技能是软件开发人员必须具备的基本技能之一，也是高等学校计算机专业毕业生应该具有的基本技能。

关系数据库技术是目前最为成熟、应用最广的数据库技术。作为关系数据库标准语言的 SQL 语言在关系数据库应用开发中起着核心的作用，熟练掌握 SQL 语言是进行关系数据库应用开发的基本前提。如何在开发工具中连接和操作关系数据库是关系数据库应用开发的另一个基本问题。本书主要针对 SQL 语言以及如何向关系数据库提交 SQL 操作这两个基本问题展开论述，目的是使读者掌握 SQL 语言的基本应用方法和技巧，以及前台开发工具连接和操作后台关系数据库的基本方法。

本书共 8 章。第 1 章简要介绍数据库、关系数据库及 SQL 语言的基本概念。第 2 章主要讨论标准 SQL 语言（ANSI 标准 SQL-99，ISO 标准 SQL3）。第 3 章给出 MS SQL Server 及 Sybase 所支持的 Transact-SQL 语言（T-SQL 语言）。第 4 章阐述 ORACLE 数据库所支持的 PL/SQL 语言。第 3 章和第 4 章内容均是围绕相应 SQL 语言与标准的差异展开论述的，并不是简单地重复相关内容。第 5 章介绍关系数据库设计的基本概念和基本方法。第 6 章介绍 VB.NET、Visual C#连接和操作数据库的方法，对通过 ADO.NET 向数据库提交 SQL 语句并完成数据库操作的设计过程和设计技巧作了介绍。第 7 章讨论 PowerBuilder 连接和操作数据库的方法，对如何连接数据库、如何向数据库提交执行 SQL 语句以及 SQL 语句的构造方法作了介绍。第 8 章介绍 Java 语言连接和操作数据库的方法，对 JDBC 技术及其应用方法作了比较深入的讨论。

感谢科学出版社对本书的出版给予的支持和帮助。

限于作者的知识水平，书中难免会有不当甚至错误之处，诚请读者批评指正。

作 者

2006 年 8 月

目 录

第 1 章 数据库概述	1
1.1 数据模型	1
1.1.1 概念模型	1
1.1.2 层次模型	2
1.1.3 网状模型	4
1.1.4 关系模型	4
1.1.5 面向对象模型	6
1.2 关系数据库管理系统及关系数据库应用	6
1.2.1 关系数据库的三级模式	6
1.2.2 关系数据库应用模式	7
1.2.3 关系数据库应用开发环境	8
1.2.4 桌面数据库管理系统	9
1.3 SQL 语言概述	9
1.4 示例数据库说明与 SQL 语句语法描述约定	10
习 题	15
第 2 章 SQL 语言	16
2.1 数据类型、关键字、标识符、注释	17
2.2 数据库和基本表定义	18
2.2.1 建立和卸载数据库	18
2.2.2 建立、修改和删除基本表	19
2.3 插入、修改与删除数据	25
2.4 查询数据	27
2.4.1 单表查询与标准函数	28
2.4.2 检索条件 (WHERE 子句)	33
2.4.3 数据分组 (GROUP BY 子句和 HAVING 子句)	38
2.4.4 输出排序 (ORDER BY 子句)	40
2.4.5 多表连接查询	41
2.4.6 子查询	47
2.5 自定义例程与永久存储模块	51

2.5.1	自定义函数与自定义存储过程	52
2.5.2	变量与控制结构	53
2.5.3	自定义例程举例	55
2.6	事务与并发控制	57
2.6.1	数据一致性问题与事务	57
2.6.2	事务的并发控制	61
2.7	触发器 (TRIGGER)	64
2.8	视图 (VIEW)	67
2.9	游标 (CURSOR) 的使用	70
2.10	索引	72
2.11	安全控制	73
2.11.1	角色 (ROLE) 与用户	73
2.11.2	操作授权	73
2.11.3	取消操作授权	74
习 题		75
第 3 章	T-SQL 语言	77
3.1	MS SQL Server	77
3.2	数据库、数据库用户、系统存储过程	79
3.3	数据类型	83
3.4	数据库操作	85
3.4.1	建立数据库对象	85
3.4.2	查询语句	98
3.5	系统变量与系统配置	107
3.5.1	系统变量	107
3.5.2	系统配置	109
3.6	系统函数	115
3.7	变量、流程控制	120
3.8	用户操作授权	124
习 题		127
第 4 章	PL/SQL 语言	128
4.1	ORACLE 9i 及其 SQL 调试环境	128
4.2	注释、模块、变量与常量说明、赋值、变量属性	132
4.3	数据类型及其相互转换	134

4.4 数据库操作.....	139
4.4.1 建立数据库对象.....	139
4.4.2 查询语句.....	146
4.5 函 数.....	149
4.6 控制结构.....	153
4.6.1 IF 语句.....	153
4.6.2 CASE 语句.....	154
4.6.3 循环及相关语句.....	156
4.6.4 NULL 语句、标号与转移.....	158
4.7 游 标.....	158
4.7.1 游标操作语句与游标属性.....	159
4.7.2 游标 FOR 循环.....	160
4.8 事务处理.....	161
4.9 意外事件处理.....	163
习 题.....	165
第 5 章 关系数据库设计.....	166
5.1 关系数据库设计概述.....	166
5.1.1 关系数据库设计步骤.....	166
5.1.2 需求分析.....	166
5.1.3 概念结构设计与 E-R 图.....	167
5.1.4 实例数据库的需求定义与概念结构设计.....	168
5.2 关系的规范化.....	169
5.2.1 依 赖.....	169
5.2.2 关系范式.....	169
5.2.3 规范化与非规范化.....	172
5.3 关系模式设计.....	173
5.4 数据库对象的可视化设计与脚本自动生成.....	174
5.4.1 MS SQL Server 2000 数据库对象的可视化建立方法.....	175
5.4.2 ORACLE 9i 数据库对象的可视化建立方法.....	185
5.4.3 数据库脚本自动生成.....	191
5.5 触发器应用.....	194
5.6 存储过程应用.....	197
5.7 一个数据汇总数据库设计实例.....	198
5.7.1 问题描述.....	198

5.7.2 数据库表设计	199
习 题	201
第 6 章 VB.NET、Visual C# 数据库连接与 SQL 操作	202
6.1 VB.NET 和 C#开发环境	202
6.1.1 Visual Studio.NET 的主界面与应用程序项目	202
6.1.2 VB.NET 和 C# 窗体代码段	204
6.1.3 VB.NET 和 C#数据库访问机制	206
6.2 ADO.NET 简介	207
6.2.1 Connection 对象	208
6.2.2 Command 对象	209
6.2.3 DataAdapter 对象	210
6.2.4 DataReader 对象	211
6.2.5 DataSet 对象	211
6.3 数据库连接与操作	212
6.3.1 在 VB.NET 中使用数据控件	212
6.3.2 在 C# 中使用数据控件	216
6.3.3 在 VB.NET 中使用 ADO.NET 对象编程	217
6.3.4 在 C#中使用 ADO.NET 对象编程	219
6.4 参数化 SQL 与 SQL 语句动态生成	221
6.4.1 参数化 SQL	221
6.4.2 SQL 语句动态生成	224
习 题	225
第 7 章 PowerBuilder 数据库连接与 SQL 操作	226
7.1 PowerBuilder 9.0 开发环境介绍	226
7.1.1 PowerBuilder 9.0 主界面、工作区、目标程序与应用程序	226
7.1.2 应用程序框架及其事件脚本	227
7.2 PowerBuilder 9.0 数据库连接方法	228
7.2.1 事务对象	229
7.2.2 连接 MS SQL Server 数据库	230
7.2.3 连接 Oracle 数据库	232
7.2.4 数据库操作与断开连接	233
7.3 嵌入式 SQL 语句应用	234
7.3.1 单行检索语句 SELECT 和 SELECTBLOB	234

7.3.2 多行检索——游标	235
7.3.3 数据库更新语句与事务处理	236
7.3.4 存储过程	239
7.3.5 动态 SQL 语句	241
7.4 数据窗口与数据存储	246
习 题	248
第 8 章 Java 数据库连接与 SQL 操作	250
8.1 Java 与 JDBC	250
8.1.1 Java 语言	250
8.1.2 JDBC 技术	251
8.2 常用 JDBC API 介绍	253
8.2.1 常用 JDBC API 简介	253
8.2.2 Connection 接口对象	253
8.2.3 DriverManager 类和 DataSource 接口对象	255
8.2.4 Statement 及其派生接口对象	256
8.2.5 ResultSet 和 RowSet 接口对象	258
8.2.6 SQLException 和 BatchUpdateException 异常处理类	261
8.3 JDBC 数据库连接与操作	262
8.3.1 通过 JDBC-ODBC 桥连接数据库	262
8.3.2 通过 MS SQL Server 2000 专用 JDBC 驱动程序连接数据库	264
8.3.3 通过 jTDS 专用 JDBC 驱动程序连接数据库	269
习 题	270
附录 ODBC 数据源设置	271
参考文献	275

第 1 章 数据库概述

数据库是长期存储在计算机内的、有组织的、可共享的数据集合^[1]。数据库是管理信息系统、企业资源计划系统以及电子商务系统等的基础和核心，绝大多数数据管理软件都离不开数据库的支持。

从技术角度讲，数据库是一组相关数据对象的定义及其数据实例的集合，依据其数据对象的数据模型的不同，数据库可以分为网状数据库、层次数据库、关系数据库和面向对象数据库等不同的类型。

1.1 数据模型

数据模型是数据库系统的基础，它决定了数据库存储和处理数据的方式。数据模型的核心是其数据结构，依据数据库系统数据存储结构的不同，数据模型分为层次模型、网状模型、关系模型和面向对象模型。

在数据库中，把具有相同结构类型的数据元素的集合作为一个逻辑整体进行管理，其中的每一个数据元素叫做一个记录。例如，图 1.1 为一个学生记录结构，图 1.2 则是对应该结构的学生记录。

学号	姓名	性别	出生日期	籍贯	所在系	所在班级
----	----	----	------	----	-----	------

图 1.1 学生记录结构

19980001	王伟	男	1978 年 1 月 1 日	吉林	计算机系	计 9801
19980002	徐大力	男	1977 年 12 月 23 日	黑龙江	计算机系	计 9801
.....						
19990001	高清	女	1978 年 12 月 29 日	上海	电子工程系	电 9902

图 1.2 学生记录表

1.1.1 概念模型

概念模型是人们对现实世界中客观事物的逻辑描述，借助于概念模型可以使人们了解数据库应用需求的基本规格，从而将其转换为数据库中的具体数据结构

并实现相应的数据管理要求。概念模型通过实体及其属性和联系来描述客观事物。

实体 (Entity) 是指客观存在的、可以相互区分的事物。例如, 高等学校的系部、班级、学生、课程、教师等都是实体。

实体所具有的某一特性称为该实体的属性 (Attribute)。一个实体一般都有多个属性。例如, 学生实体可以包含学号、姓名、性别等属性。

实体之间是可以相互区分的。区分实体的依据一般是一个属性或属性组, 这个可以唯一标识一个实体的属性或属性组叫做码 (Key)。

一般而言, 属性的取值不是随意的, 任一属性的取值都有一定的范围, 如学生的出生日期应由合法的年份、月份和日组成。属性的取值范围叫做该属性的域 (Domain)。

实体一般不是孤立的, 它们之间往往存在某种对应关系, 这种对应关系叫做联系 (Relationship)。实体之间的联系分为如下三种:

第一种联系是一对一联系。例如, 班级实体和班主任实体之间就是一对一的联系, 即一个班只有一个班主任、一个班主任只带一个班。一对一联系一般用 $1:1$ 表示。

第二种联系是一对多的联系。比如, 班级实体和学生实体之间就存在一对多的联系, 即一个班中可以有多个学生。一对多联系一般用 $1:n$ 表示。

第三种联系是多对多联系。例如, 学生实体和课程实体之间就是多对多联系, 即一门课程可以同时有多个学生选修, 一个学生可以同时选修多门课程。多对多联系一般用 $m:n$ 表示。

在表示实体及其联系时, 一般用实体类型, 即实体的型来描述。具有相同类型的实体的集合叫做实体集, 其中的任何一个实体都是该实体型的一个实例。

概念模型的常用表示方法是 E-R (Entity-Relationship) 图, 其组成要素如下:

- (1) 实体型用矩形框表示, 矩形框内标注实体名。
- (2) 属性用椭圆形表示, 椭圆内标注属性名, 用直线连接实体型及其属性。
- (3) 联系用菱形表示, 菱形内标注联系名, 用直线连接联系及相关实体, 同时在连线上标注联系的类型。

图 1.3 描述的是一个学籍管理系统的 E-R 图。

E-R 图是对客观事物的逻辑抽象。借助 E-R 图可以直观地了解数据库应用需求并进而设计出满足该需求的数据库系统。有时, 为了简化 E-R 图, 可以只在 E-R 图中画出实体及其相互联系, 实体属性则单独给出。

1.1.2 层次模型

层次模型是早期的数据库系统所采用的数据模型, 如 IBM 公司的 IMS 数据库即是一个典型的代表。

层次模型以树型结构组织数据, 树的每一个结点表示一种记录类型。例如,

假定学校中系之下设教研室和班级，则可定义如图 1.4 所示的系部、教研室、班级层次结构。图 1.5 是对应于图 1.4 的数据实例。

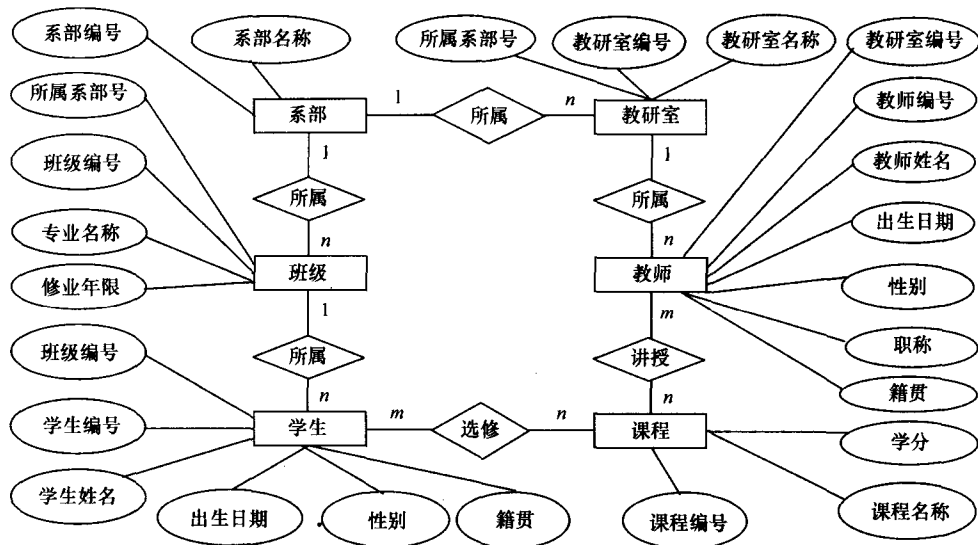


图 1.3 学籍管理系统的 E-R 图

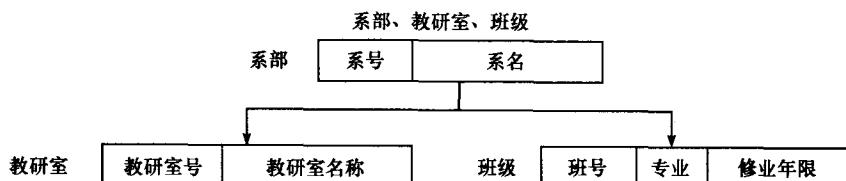


图 1.4 教研室、班级层次结构

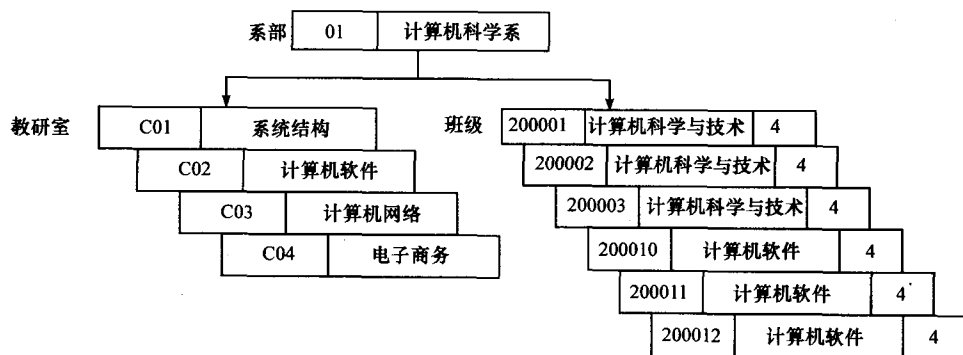


图 1.5 教研室、班级层次结构的一个数据实例

层次数据模型的数据结点须满足以下两点要求:

- (1) 有且只有一个结点没有双亲结点, 该结点称为根结点。
- (2) 除根结点之外的其他所有结点均有且只有一个父结点。

层次模型比较简单和直观, 对于一对多的联系进行描述比较方便。

1.1.3 网状模型

网状模型是在 20 世纪 70 年代由 CODASYL (Conference On Data System Language) 下属的 DBTG (数据库任务组, Data Base Task Group) 提出的。典型的网状模型数据库有 Cullinet 公司的 IDMS、Univac 公司的 DMS1100 和 HP 公司的 IMAGE 等。

网状模型是层次模型的扩充, 它以图的结构组织数据, 图的每一个结点表示一种记录类型。网状数据模型的数据结点具有以下特点:

- (1) 可以有一个以上的结点没有双亲结点。
- (2) 一个结点可以有一个或多个双亲结点。

网状模型可以灵活地描述实体及其联系, 同时, 网状模型的性能也较高。

1.1.4 关系模型

关系模型是目前大多数数据库系统所采用的数据模型。关系模型最早由 IBM 公司的 E.F. Codd 博士于 1970 年提出并建立了完整的理论体系。

层次模型和网状模型相对而言实现都比较困难, 同时又存在一些局限性。在关系模型提出之后, 由于其理论完备、概念单一、易于实现, 所以受到特别的重视并很快开发出基于关系模型的数据库软件。关系数据库的典型代表有 ORACLE 数据库、Informix 数据库、Sybase 数据库和 MS SQL Server 等。

1. 关系的基本概念

关系模型的核心是关系 (Relation)。简单地说, 关系就是一个二维表。图 1.6、图 1.7 所示的就是两个典型的关系。

部门号	部门名
01	计算机科学系
02	电子工程系
03	管理科学系
04	数理系

图 1.6 关系表

关系 (表) 中的一行叫做一个元组 (Tuple)。元组也可以称为记录。

关系 (表) 中的每一列叫做一个属性 (Attribute)。元组中与属性对应的取值称为分量或数据项。

关系中可以唯一标识一个元组的属性或属性组可以作为该关系的主键 (Primary Key)。例如, 图 1.6 中的部门号可以作为该关系的主键, 图 1.7 中的所属部门号和教师号可以作为该关系的主键 (这种包含多个属性的主键一般也叫复合主键)。

所属部门号	教师号	教师名	性别	出生日期	籍贯	学历
01	001	王大伟	男	1972 年 1 月 1 日	吉林	大学本科
01	002	周婷婷	女	1969 年 10 月 1 日	黑龙江	大学专科
02	001	徐德林	男	1970 年 5 月 12 日	北京	硕士
...	...	...	...	...	...	...

图 1.7 关系表

2. 关系模式

关系是元组的集合。关系及其属性的形式化定义叫做关系模式，其定义形式如下：

关系名 (属性名 1, 属性名 2, ..., 属性名 n)

图 1.6、图 1.7 的关系模式可以如下描述：

部门 (部门号, 部门名)

教师 (所属部门号, 教师号, 教师名, 性别, 出生日期, 籍贯, 学历)

对关系模型的基本要求是其属性必须是原子属性，即该属性不可再分。在关系模式中，一般把关系的主键属性或属性组用下划线标出（部门关系中的“部门号”，教师关系中的“所属部门号”和“教师号”）。

3. 关系数据的完整性

数据的完整性是指数据的正确性、有效性和相容性。关系数据的完整性包括三个方面：一是实体完整性，二是参照完整性，三是用户定义完整性。

实体完整性是指关系的主键不得为空且取值必须唯一，否则，相应记录（元组）不能被输入到关系表中。实体完整性保证了关系中不会含有不确定或不可区分的实体。

参照完整性是指当一个表中包含另外一个表的主键属性（或属性组）时，该关系在相应属性或属性组上的取值或者为空，或者为另一个表中已有的分量值。例如，教师关系（参见图 1.7）中的所属部门号只能取 01、02、03、04 或空值，如果取 08 则表示相应教师属于一个不存在的部门，这是不允许的。

实体完整性和参照完整性可以由关系数据库软件自动维护，也可以通过编程进行维护。

用户定义完整性由用户根据实际应用进行定义，以保证数据库数据的正确性。例如，教师关系中的性别属性一般只能在“男”和“女”之间取值，出生日期不应该超出某一合理的范围，籍贯不能是一个不存在的省份等都属于用户定义完整性的范畴。

1.1.5 面向对象模型

面向对象模型是面向对象程序设计方法学与数据库技术结合的产物^[1]。面向对象模型用对象来描述实体及其相互联系，每一个实体被抽象为一个对象。

1.2 关系数据库管理系统及关系数据库应用

数据库数据模型是由数据库管理系统 (Database Management System, DBMS) 实现的。数据库管理系统是用于建立、维护和操作数据库的软件集合。依据数据库管理系统所支持的数据模型的不同，数据库管理系统可以分为层次、网状、关系和面向对象数据库管理系统。关系数据库管理系统是目前应用最广泛、技术最成熟的数据库管理系统。

1.2.1 关系数据库的三级模式

关系数据库管理系统一般都支持三级模式 (Schema) 结构 (参见图 1.8)。

用户子模式是用户应用程序操作数据库的接口，是用户所看到的数据视图，它一般与实际的数据库模式有很大的差别。

数据库模式是数据库中所有关系模式描述的集合，即数据结构描述的集合，它是所有数据库用户的公共数据视图。数据库设计的核心问题是数据库模式的设计。

存储模式定义了数据库数据的具体存储方式，它决定了数据库数据在外存设备上的物理存储结构。

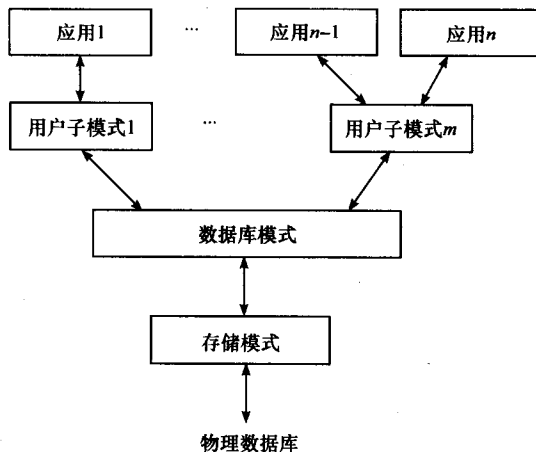


图 1.8 关系数据库的三级模式结构

关系数据库管理系统通过数据描述语言 (Data Definition Language, DDL) 描述数据库的用户子模式、数据库模式和存储模式。

关系数据库管理系统提供了必要的操作手段以控制实现由数据库模式到用户子模式和存储模式的映像。

1.2.2 关系数据库应用模式

关系数据库应用模式可以分为多用户模式、客户/服务器模式、三层/多层应用模式和分布式应用模式。

早期的关系数据库都采用多用户应用模式。多用户模式下的数据库管理系统和客户应用程序运行在同一主机上并直接进行交互, 用户通过终端运行其应用程序。多用户应用模式对主机的性能要求比较高。

客户/服务器应用模式是多用户应用模式在局域网络环境下的一种实现方式, 在此模式下, 数据库管理系统作为服务器响应客户应用程序的数据操作请求并将操作结果及状态信息返回给客户应用程序, 客户应用程序对返回的数据进行必要的处理后再传回后台数据库或输出给用户。在客户/服务器应用模式下, 数据库管理系统和客户应用程序一般分别运行在网络上的不同计算机上 (也可以运行在同一计算机上), 数据库管理系统所运行的计算机一般仍需具有较高的性能。

多层数据库应用模式是客户/服务器模式的多级实现形式, 其典型结构是三层应用模式。三层应用模式包括数据源层 (数据库服务器)、中间层 (业务逻辑应用服务器) 和客户端 (Web 浏览器或前台客户应用)。

数据库服务器运行 DBMS, 它响应中间层应用的数据请求。

中间层封装组织业务规则的组件, 它从数据源提取并加工数据, 然后把处理结果传递到客户端, 或者响应客户端的数据库操作请求并在进行必要数据处理之后更新到后台数据库。中间层隔离了前台客户端与后台数据库的直接联系, 从而保证了后台数据库的安全。中间层的存在也便于对软件的维护和管理。

客户端为用户操作接口, 它可以是浏览器或客户应用程序, 它仅仅完成操作请求和数据展示的功能。在修改软件功能时, 几乎不需要修改客户端软件, 而只是修改中间层应用服务组件即可。基于 Web 的三层应用模式是目前最典型的三层应用模式。

在分布式应用模式下, 数据库源数据分布在网络上的不同数据库上, 甚至一个数据库的数据分布在网络上的不同数据库服务器上。在分布式应用模式下, 数据库操作可能会涉及不同数据存储点的数据, 即操作是分布的。支分布式应用的数据库系统会提供相应的机制以管理分布式数据库操作。

1.2.3 关系数据库应用开发环境

开发数据库应用程序的第一步是选择一种数据库管理系统并安装配置一个数据库服务器。数据库服务器是一个 DBMS 命名安装实例，它一般安装在网络上的一台专用的、高性能的计算机上。在大型应用系统中，可能需要安装多个数据库服务器。数据库服务器的操作系统环境既可选择 Windows（比如 Windows 2000 Server），也可以选择 UNIX 或 Linux。数据库是数据库服务器上命名的表、视图、存储过程等相关数据库对象的集合。数据库从属于某一个数据库服务器。在任意一个数据库服务器上都可以有多个数据库，例如，在 Sybase 或 MS SQL Server 服务器中就有诸如 master、model 等系统数据库。

用户要操作数据库，首先必须能够访问数据库所在的数据库服务器，同时，还必须具有访问相应数据库的资格。用户要能够访问数据库服务器，就必须拥有相应数据库服务器的登录账号。数据库服务器中的登录账号具有不同的权限，以 MS SQL Server 为例，sa 是初始安装后的系统管理员账号，它拥有对整个数据库服务器的所有操作权限。一般用户应使用专用账号，其权限应由系统管理员作出严格的限制以保证系统的安全。如果一个服务器登录账号具有建立数据库的权限，它就可以在服务器上建立数据库，而建立这个数据库的账号就成为相应数据库的所有者。任一登录账号要想访问某一数据库，必须首先成为该数据库的合法用户，这可由系统管理员来授权。同一数据库的不同用户拥有的数据库操作权限一般是不同的。数据库所有者拥有对其所拥有的数据库的全部操作权限并可为其他用户进行操作授权。获得建立数据库对象权限的用户可以建立相应数据库对象并成为该对象的所有者，其他用户必须获得所有者的授权才能访问或使用相应数据库对象。数据库管理的通行做法是由系统管理员建立数据库和数据库对象，然后将相关数据库登录账号增加为指定数据库的用户并对其进行相应的操作授权。

要开发数据库应用程序，还必须选择一种合适的应用程序开发工具。目前常用的数据库应用开发工具有 Visual Basic、Delphi 和 PowerBuilder 等。开发工具提供了多种操作数据库的控件和系统对象以及界面设计元素，利用这些编程对象可以设计出满足各种应用需求的数据库应用程序。

习惯上，人们把数据库服务器上的数据库叫做后台数据库，而把开发工具叫做前台开发工具。前台开发工具要能访问后台数据库，就必须能够寻址并连接后台数据库。常用的数据库连接方式有 ODBC 方式和 OLE DB 方式。

ODBC（Open Database Connectivity）接口是一种标准化的接口，它能让前台应用程序在各种数据库服务器上存取数据。ODBC 通过标准的 SQL 语言与后台数据库交互。