

Oracle数据库 应用与开发

石彦芳 李丹 主编



Oracle Database Application
and Development

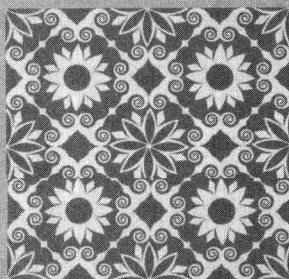


计算机应用技术规划教材

Oracle数据库 应用与开发

石彦芳 李丹 主编
赵占坤 赵滨 薛玉倩 副主编
周檬 苏默 高秀艳 石建国 参编

Oracle Database Application
and Development



机械工业出版社
China Machine Press

本书是作者在多年从事数据库教学和开发的基础上编写而成的，书中全面介绍了 Oracle 数据库应用和开发的知识，内容涵盖关系型数据库、Oracle 数据库的体系结构、用户与方案的创建与管理、表的创建与管理、SQL 基本查询、SELECT 高级查询、PL/SQL 编程基础、存储过程与函数的创建、索引与视图的创建、序列和同义词的创建、触发器的创建与应用、事务处理与并发控制、数据库安全性管理、数据库的备份与恢复以及数据库综合实训。通过本书的学习，读者能够快速掌握 Oracle 的相关知识并进行数据库的开发。

本书适合作为普通高校计算机及相关专业数据库应用的教材，也可供数据库开发和应用人员参考阅读。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目 (CIP) 数据

Oracle 数据库应用与开发 / 石彦芳, 李丹主编. —北京: 机械工业出版社, 2012. 3

(计算机应用技术规划教材)

ISBN 978-7-111-37463-3

I. O… II. ①石… ②李… III. 关系数据库-数据库管理系统, Oracle -高等学校-教材 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2012) 第 022418 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 刘立卿

中国电影出版社印刷厂印刷

2012 年 3 月第 1 版第 1 次印刷

185mm × 260mm · 17.75 印张

标准书号: ISBN 978-7-111-37463-3

定价: 35.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzsj@hzbook.com

前 言

Oracle 数据库是当今应用广泛的关系型数据库系统，它由 Oracle 公司 1977 年推出，是 Oracle 公司的核心产品，至今已有 30 余年的发展历史，目前在市场上占有主要份额。作为一种大型网络数据库管理系统，Oracle 数据库功能强大，能够处理大批量数据，主要应用于商业和政府部门。

Oracle 10g 是 Oracle 公司在 2004 年 2 月正式发布的，从这一版开始，Oracle 数据库有了一个新的后缀 g (g 即 grid, 网格)，主打网格计算。本书以 Windows XP 操作系统为平台，基于 Oracle 10g 版本，由浅入深地介绍了 Oracle 数据库系统的使用方法、标准 SQL 语言的应用、Oracle 数据库专用编程语言 PL/SQL 的应用、数据库的备份和恢复知识。在本书的最后还以一个基于 Web 的音乐网站项目的开发为案例，综合介绍了数据库的分析、设计、实现和使用方法。

本书是作者在多年从事数据库教学和开发的基础上编写而成的，采取理论和实践相结合的方式，全面介绍了 Oracle 数据库应用和开发的知识，使读者在学习之后能够快速掌握 Oracle 的相关知识并进行数据库的开发。

本书适合作为普通高校计算机及其相关专业数据库应用的教材，也可以作为 Oracle 数据库开发和应用人员的参考资料。

本书由石彦芳、李丹担任主编，赵占坤、赵滨、薛玉倩担任副主编，周檬、苏默、高秀艳、石建国参编。其中，第 1 章由石彦芳编写，第 2 章由苏默编写，第 3、7、8 章由薛玉倩编写，第 4、5、6 章由李丹编写，第 9、10、11 章由赵滨编写，第 12 章由周檬编写，第 13、14、15 章由赵占坤编写，本书所有章节的实验和习题由高秀艳和石建国编写。全书的整理和审稿工作由石彦芳、李丹负责。

在编写过程中，参考了大量的相关技术资料 and 程序开发源码资料，在此向资料的作者深表谢意。书中全部程序都已上机调试通过。由于作者水平和时间有限，书中难免有错误和疏漏之处，敬请各位同行和读者不吝赐教，以便及时修订和补充。

另外，如果读者在使用本书的过程中有什么问题可直接与作者联系，E-mail: shiyanfang_123@163.com。

教学建议

本书共分 15 章，主要内容和教学建议如下：

教学内容	学习要点及教学要求	课时安排
第 1 章 关系型数据库及 Oracle 10g 介绍	本章介绍了关系型数据库和 Oracle 数据库的基础知识，为以后的学习奠定基础。主要内容包括：关系型数据库的相关概念、数据库设计的 E-R 模型、数据库范式的应用；Oracle 数据库的发展过程；Oracle 10g 数据库的安装、主要管理工具的使用方法；Oracle 数据库的启动与关闭。	4~5
第 2 章 Oracle 数据库的体系结构	本章介绍了 Oracle 数据库的体系结构，这有助于读者了解 Oracle 数据库的整体结构并有助于数据库的维护和优化工作。主要内容包括：物理存储结构的介绍；逻辑存储结构的介绍；数据库实例结构的介绍；Oracle 服务器和客户端的网络配置。	4~5
第 3 章 用户、方案的创建与管理	本章介绍了 Oracle 数据库用户与方案的概念、使用方案组织数据库对象的思想以及 Oracle 中数据库用户的创建和管理。	3~4
第 4 章 表的创建与管理	本章介绍了数据表的创建、管理和使用。表作为关系型数据库中存储数据的基本单元，是用户操作数据库的基础。主要内容包括：表结构的创建、修改和删除；根据业务规则为表添加各种约束；对表中的数据执行插入、更新与删除操作。	5~6
第 5 章 SQL 基本查询	本章介绍了关系型数据库中最常用的 SQL 语句——SELECT 查询语句的基本语法。主要内容包括：SELECT 语句及其各子句的作用与格式；SQL * Plus 中经常使用的函数；在 SQL * Plus 中格式化查询结果；SQL 脚本文件的创建与执行。	5~6
第 6 章 SELECT 高级查询	本章介绍了 SELECT 查询语句的高级应用。这部分内容在实际数据处理中应用非常广泛。主要内容包括：在 SELECT 语句中执行多个表的连接查询；SELECT 子查询在各种 DML 命令中的使用；查询结果的集合操作。	5~6
第 7 章 PL/SQL 编程基础	本章介绍了 Oracle 数据库专用的编程语言 PL/SQL 的基础知识。主要内容包括：PL/SQL 程序块的基本结构；常见 SQL 语句在 PL/SQL 程序块中的使用规则；PL/SQL 中的数据类型和流程控制语句；游标和异常的应用。	5~6
第 8 章 存储过程与函数的创建	本章主要介绍了 PL/SQL 中命名程序块的创建、调用和管理。命名程序块包括被永久存储在数据库中的过程和函数。	4
第 9 章 索引与视图的创建	本章主要介绍了索引和视图两种方案对象的概念、优点、创建和使用，以及利用视图更新数据源表的方法。	3~4
第 10 章 序列和同义词的创建	本章主要介绍了序列和同义词的概念及作用；序列的创建、应用及管理；同义词的创建、应用及管理。	2
第 11 章 触发器的创建与应用	本章主要介绍了触发器的概念及组成；触发器的类型；创建各种类型的触发器的语法格式；触发器的应用；管理现有触发器的各种操作。	3~4
第 12 章 事务处理与并发控制	本章介绍了事务处理的相关知识。事务作为数据库操作中的基本单元，有效控制着数据的处理和并发操作。主要内容包括：事务的概念与特性；常见事务管理命令；并发控制与锁。	4

(续)

教学内容	学习要点及教学要求	课时安排
第 13 章 数据库安全性管理	本章主要介绍了 Oracle 数据库中安全性管理的相关措施。数据的安全性管理是任何一种数据库都要解决的问题。在 Oracle 数据库中主要利用权限、用户、角色、概要文件进行安全性的管理和控制。	4~5
第 14 章 数据库的备份与恢复	本章主要介绍了数据库备份与恢复的概念；数据库备份的种类与策略；数据库的脱机冷备份和联机热备份；数据库的导出/导入操作。	4
第 15 章 数据库综合实训	本章主要通过一个基于 Web 的音乐网站项目，综合叙述了 Oracle 数据库在实际项目中的分析、设计、实现和使用过程。	5~6
实验和习题课	本书各章最后都附有实验和习题，主要针对每章的重点和难点知识进行练习。任课教师可以根据学生掌握的情况，要求学生选做一些习题来巩固课堂知识。如有时间，也可以安排 2 至 3 次习题课。在习题课上可以由教师讲解以前课外作业中存在的带有普遍性的问题，也可以安排稍难一些的习题让学生在课上做出解答，然后由教师指导进行讨论，最后评出最优的的答案。	4~6

目 录

前言

教学建议

第 1 章 关系型数据库及 Oracle 10g

介绍 1

1.1 关系型数据库概述 1

1.1.1 关系型数据库的相关概念 2

1.1.2 SQL 命令概述 5

1.2 数据库的设计 6

1.2.1 数据库设计的 E-R 模型 6

1.2.2 利用 E-R 模型设计表结构 8

1.2.3 数据库设计的范式理论 9

1.3 Oracle 数据库概述 11

1.3.1 Oracle 数据库的发展 11

1.3.2 Oracle 数据库的系统结构 11

1.3.3 Oracle 10g 介绍 13

1.4 Oracle 10g 的安装 14

1.4.1 安装 Oracle 10g 的软硬件要求 14

1.4.2 安装 Oracle 10g 14

1.4.3 检验安装是否成功 23

1.4.4 Oracle 数据库的默认用户 25

1.5 Oracle 系统管理工具介绍 26

1.5.1 数据库配置助手 26

1.5.2 Oracle 企业管理器 (OEM) 26

1.5.3 SQL * Plus 与 iSQL * Plus 30

1.5.4 网络配置助手 33

1.6 Oracle 数据库的启动与关闭 34

1.6.1 使用命令启动与关闭数据库 34

1.6.2 使用 OEM 工具启动与关闭

数据库 38

1.6.3 开机后自动启动与关闭数据库 39

1.7 实验 39

1.8 习题 40

第 2 章 Oracle 数据库的体系结构 41

2.1 物理存储结构 41

2.1.1 数据文件 41

2.1.2 日志文件 43

2.1.3 控制文件 45

2.1.4 参数文件 45

2.2 逻辑存储结构 46

2.2.1 表空间 47

2.2.2 段 48

2.2.3 盘区 48

2.2.4 数据块 49

2.3 数据库实例结构 50

2.3.1 进程结构 50

2.3.2 内存结构 54

2.4 Oracle 网络配置 55

2.4.1 客户端配置 55

2.4.2 服务器端配置 57

2.5 实验 57

2.6 习题 57

第 3 章 用户、方案的创建与管理 59

3.1 用户、方案概述 59

3.1.1 用户与方案的概念 59

3.1.2 方案对象与非方案对象 60

3.2 创建用户 60

3.2.1 使用 OEM 工具创建新用户 60

3.2.2 使用 SQL 命令创建新用户 62

3.2.3 使用新用户连接数据库 63

3.3 管理用户 64

3.3.1 使用 OEM 工具修改用户 64

3.3.2 使用 SQL 命令修改用户 66

3.3.3 启用与禁用用户 66

3.3.4 删除用户 67

3.4 实验 67

3.5 习题 67

第 4 章 表的创建与管理	69	5.3.1 SQL * Plus 环境中的常用格式化选项	109
4.1 创建表	69	5.3.2 使用“环境”对话框设置格式化选项的值	110
4.1.1 基本数据类型	69	5.3.3 使用命令设置格式化选项的值	110
4.1.2 使用 OEM 工具创建表	70	5.4 SQL 脚本文件的创建与执行	113
4.1.3 使用 SQL 命令创建表	74	5.4.1 创建 SQL 脚本文件	113
4.1.4 基于已有的表创建新表	75	5.4.2 执行 SQL 脚本文件	114
4.1.5 定义表中字段的默认值	75	5.5 实验	115
4.1.6 使用 DESCRIBE 命令查看表结构	76	5.6 习题	116
4.2 向表中插入、修改和删除数据	76	第 6 章 SELECT 高级查询	118
4.2.1 插入数据	76	6.1 简单连接查询	118
4.2.2 修改数据	77	6.1.1 两表的笛卡儿积运算	118
4.2.3 删除数据	78	6.1.2 表之间的简单连接查询	119
4.3 修改表与删除表	79	6.1.3 为表设置别名	120
4.3.1 修改表	79	6.2 使用 JOIN 关键字的连接查询	120
4.3.2 删除表	82	6.2.1 内连接查询	121
4.4 定义数据完整性	82	6.2.2 外连接查询	123
4.4.1 定义主键约束	83	6.2.3 交叉连接	125
4.4.2 定义不允许为空约束	87	6.3 SELECT 查询的集合操作	125
4.4.3 定义唯一性约束	87	6.3.1 UNION 集合运算	125
4.4.4 定义检查约束	88	6.3.2 INTERSECT 集合运算	126
4.4.5 定义外键约束	89	6.3.3 MINUS 集合运算	126
4.5 实验	91	6.4 子查询	127
4.6 习题	92	6.4.1 单行子查询	127
第 5 章 SQL 基本查询	94	6.4.2 多行子查询	128
5.1 SELECT 基本查询语句	94	6.4.3 多列子查询	130
5.1.1 SELECT 子句和 FROM 子句	94	6.4.4 相关子查询	132
5.1.2 WHERE 子句	98	6.4.5 在 FROM 子句中使用子查询	132
5.1.3 ORDER BY 子句	103	6.5 实验	133
5.1.4 使用统计函数	104	6.6 习题	134
5.1.5 GROUP BY 子句	105	第 7 章 PL/SQL 编程基础	137
5.1.6 HAVING 子句	106	7.1 PL/SQL 程序块结构	137
5.2 SQL * Plus 中常用函数介绍	107	7.2 变量与常量	139
5.2.1 字符串函数	107	7.2.1 PL/SQL 标识符	139
5.2.2 数值函数	108	7.2.2 PL/SQL 中的数据类型	140
5.2.3 日期和时间函数	108	7.2.3 声明变量与常量	141
5.2.4 转换函数	109	7.2.4 为变量和常量赋值	142
5.3 使用 SQL * Plus 命令格式化查询结果	109	7.2.5 变量和常量的作用域	142

7.3 常见 SQL 语句在 PL/SQL 程序中的使用	143	8.3 实验	175
7.3.1 SELECT 语句在 PL/SQL 程序中的使用	143	8.4 习题	177
7.3.2 INSERT、UPDATE、DELETE 语句在 PL/SQL 程序中的使用	144	第 9 章 索引与视图的创建	179
7.3.3 DCL 语句在 PL/SQL 程序中的使用	146	9.1 索引概述	179
7.3.4 在 PL/SQL 程序中间接使用 DDL 语句	147	9.1.1 索引的概念	179
7.4 PL/SQL 中的复合数据类型	147	9.1.2 索引的类型	180
7.4.1 记录类型	147	9.2 创建索引	183
7.4.2 记录表类型	148	9.2.1 创建 B 树索引	184
7.5 使用 %TYPE 和 %ROWTYPE 定义变量	149	9.2.2 创建位图索引	184
7.5.1 使用 %TYPE 定义简单变量	149	9.2.3 创建反向键索引	184
7.5.2 使用 %ROWTYPE 定义记录变量	150	9.2.4 创建基于函数的索引	185
7.6 PL/SQL 中的流程控制语句	150	9.3 应用索引	185
7.6.1 条件选择语句	150	9.4 视图概述	186
7.6.2 循环语句	154	9.4.1 视图的概念	186
7.7 游标的创建与应用	156	9.4.2 视图的优点	187
7.7.1 显式游标	156	9.5 创建与管理视图	188
7.7.2 带参数的游标	160	9.5.1 创建和应用视图	188
7.7.3 隐式游标	161	9.5.2 重新编译视图	189
7.7.4 使用游标更新表中的数据	161	9.5.3 删除视图	189
7.8 异常处理	163	9.6 创建可更新视图	189
7.8.1 PL/SQL 程序块中的异常部分	163	9.7 实验	192
7.8.2 系统异常	163	9.8 习题	194
7.8.3 用户自定义异常	165	第 10 章 序列和同义词的创建	195
7.9 实验	166	10.1 序列的创建、应用及管理	195
7.10 习题	166	10.1.1 创建序列	195
第 8 章 存储过程与函数的创建	168	10.1.2 应用序列	196
8.1 存储过程	168	10.1.3 管理序列	197
8.1.1 创建与调用存储过程	168	10.2 同义词的创建、应用及管理	197
8.1.2 修改与删除存储过程	173	10.2.1 创建、应用同义词	197
8.2 函数	173	10.2.2 管理同义词	198
8.2.1 创建与调用函数	173	10.3 实验	198
8.2.2 修改与删除函数	175	10.4 习题	199
		第 11 章 触发器的创建与应用	200
		11.1 触发器概述	200
		11.2 触发器类型	201
		11.3 创建触发器	202
		11.3.1 创建 DML 事件触发器	202
		11.3.2 创建 DDL 事件触发器	206
		11.3.3 创建替代触发器	209

11.3.4 创建用户事件触发器	211	13.4 对象权限管理	236
11.3.5 创建系统事件触发器	213	13.4.1 对象权限分类	236
11.4 管理触发器	214	13.4.2 对象权限的授权	236
11.4.1 修改触发器	214	13.4.3 对象权限的回收	237
11.4.2 删除触发器	214	13.5 角色管理	237
11.5 实验	215	13.5.1 角色概述	237
11.6 习题	216	13.5.2 系统预定义角色	238
第12章 事务处理与并发控制	217	13.5.3 用户自定义角色	238
12.1 事务概述	217	13.5.4 删除角色	239
12.2 事务的特性	218	13.6 数据库概要文件	239
12.2.1 原子性	218	13.6.1 数据库概要文件概述	239
12.2.2 一致性	221	13.6.2 创建数据库概要文件	240
12.2.3 隔离性	221	13.6.3 管理数据库概要文件	242
12.2.4 持久性	222	13.6.4 查看概要文件的信息	242
12.3 事务管理命令	223	13.7 实验	243
12.3.1 COMMIT 命令	223	13.8 习题	243
12.3.2 ROLLBACK 命令	224	第14章 数据库的备份与恢复	244
12.3.3 SAVEPOINT 和 ROLLBACK TO SAVEPOINT 命令	224	14.1 Oracle 的备份与恢复机制	244
12.3.4 SET TRANSACTION 命令	225	14.1.1 数据库备份的重要性	244
12.3.5 SET CONSTRAINT 命令	226	14.1.2 数据库备份的内容	244
12.4 并发控制与锁	227	14.1.3 数据库备份的种类	245
12.4.1 并发问题	227	14.1.4 数据库备份中的保留策略	246
12.4.2 锁	228	14.2 冷备份	246
12.5 实验	229	14.2.1 冷备份概述	246
12.6 习题	230	14.2.2 冷备份操作步骤	247
第13章 数据库安全性管理	231	14.2.3 冷备份恢复步骤	249
13.1 Oracle 数据库安全性管理概述	231	14.3 热备份	250
13.1.1 企业信息系统安全问题	231	14.3.1 热备份概述	250
13.1.2 Oracle 信息安全整体解决 方案	231	14.3.2 热备份操作步骤	250
13.1.3 Oracle 数据库安全策略	232	14.4 EXP/IMP 逻辑备份	251
13.1.4 Oracle 数据库访问的身份 验证	233	14.4.1 EXP 导出数据	252
13.2 权限概述	233	14.4.2 IMP 导入数据	254
13.3 系统权限管理	234	14.5 实验	255
13.3.1 系统权限分类	234	14.6 习题	255
13.3.2 系统权限的授权	234	第15章 数据库综合实训	256
13.3.3 系统权限的回收	235	15.1 系统设计	256
		15.1.1 系统功能概述	256
		15.1.2 系统功能模块设计	256
		15.2 数据库设计	257

15.2.1	数据库需求分析	257	15.4.2	使用 JDBC 技术连接 Oracle 数据库	263
15.2.2	数据库逻辑结构设计	257	15.4.3	访问数据库的工具类的实现 ...	265
15.3	数据库实现	259	15.4.4	核心模块的实现	269
15.3.1	创建 musicSite 用户	259	15.4.5	帮助模块的实现	271
15.3.2	创建表和约束	259	15.5	系统的编译、打包和发行	271
15.4	项目实施	263			
15.4.1	创建 Web 项目——musicSite ...	263			

第 1 章 关系型数据库及 Oracle 10g 介绍

关系型数据库是当今应用最广泛的数据库，本章主要介绍关系型数据库的相关概念和概述 Oracle 10g 数据库。关系型数据库以二维表为基本数据存储形式，利用 SQL 语句实现对数据的各种操作。E-R 模型是数据库分析与设计的辅助工具，可以使数据库的设计更合理。Oracle 10g 是网络关系型数据库，应用广泛，本章将介绍它的安装过程、主要管理工具、启动与关闭以及连接的基础知识。

本章要点

- 关系型数据库的概念及设计模型
- Oracle 10g 的介绍与安装
- Oracle 10g 的主要管理工具
- Oracle 数据库的启动与关闭

1.1 关系型数据库概述

数据是重要的资源，在生产和生活中起着重要的作用，尤其是全球已步入了信息化时代，各行各业更是离不开数据。数据是用来描述客观事物属性的重要手段，能够表达事物外在或内在的特征。比如，学生有学号、姓名、性别、生日、所在班级等属性；课程有课程编号、课程名称、课时、学分等属性。将这些属性表达并记录下来就成为了数据。当然，也可以用数据描述已发生事件的特征和结果。比如，考试是个事件，我们需要记录每位学生每门课的考试成绩和考试日期，这些也是数据。类似的例子还有很多，我们经常会为了生活或工作的需要收集、存储和使用数据。

既然数据对我们来说如此重要，那么如何收集、分析、存储和加工数据就值得我们关注，因为这些技术或方法直接影响着我们对数据的使用效率和效果。数据管理大致经历了三个阶段：人工管理阶段、文件管理阶段和数据库管理阶段。

1. 人工管理阶段

20 世纪 50 年代中期以前，计算机主要用于科学计算。当时没有磁盘，没有操作系统和管理数据的软件，数据的存储介质主要是纸，如各种档案和书面文件。这种存储介质容易损坏，不易长期保存，不能被有效地管理，数据是面向程序的而不具有独立性，更不能共享。

2. 文件管理阶段

20 世纪 50 年代后期到 60 年代中期，计算机硬件和软件技术得到了发展，这时有有了磁盘，操作系统中已经有了专门的数据管理软件——一般称为文件系统。早期的文件系统将数据和程序存储在一起，如 BASIC 语言；后来发展到数据和程序单独存储，用程序文件调用数据文件，二者实现了相对独立，如 C 语言。但是，当时的数据文件主要是受操作系统管理，数据管理功能较少，主要实现数据的存储和简单查询，不能实现更多复杂的数据操作。此阶段的特点是：数据文件可以长期保存、数据冗余度大、程序与数据依赖性强、数据处理操作单一并且数据之间没有联系、数据缺乏统一的管理和控制。

3. 数据库管理阶段

20 世纪 60 年代后期,随着计算机硬件和软件技术的飞速发展,出现了数据库技术,它能更有效地管理和存取大量的数据,避免以上两个阶段的缺点。采用数据库技术可实现数据共享、减少数据冗余、提高数据独立性,能够进行统一的数据管理和控制。

数据库管理技术的发展大致经历了三个阶段:初级阶段、中级阶段和高级阶段。

- 初级阶段:主要以层次型数据库和网状数据库的应用为主。当时 IBM 公司的 IMS (Information Management System, 信息管理系统) 层次模型数据库发展很快,客户群巨大,直到现在该系统仍在使用和不断发展。网状数据库系统相对来说更复杂也更具有专用性,因此没有得到广泛的应用。
- 中级阶段:主要以关系型数据库的发展与应用为主。1970 年,IBM 公司的 E. F. Codd 博士提出了关系型数据库模式,为以后关系型数据库的发展奠定了理论基础。相比以前的层次数据库和网状数据库,关系型数据库实现起来更简单,因此它很快发展起来。20 世纪 80 年代初期,IBM 公司的关系型数据库系统 DB2 问世。而当时的 Oracle 公司是一家专门做关系型数据库的公司,它将 Oracle 数据库移植到桌面计算机上,成为占主导地位的主流数据库,并逐渐取代了层次与网状数据库。关系型数据库产品很多,如桌面关系数据库管理系统有 FoxBase、Visual Foxpro、Access 等;大型关系数据库管理系统有 SQL Server、DB2、Oracle、Sybase、Informix 等。
- 高级阶段:计算机的应用从传统的科学计算、事务处理等领域逐步扩展到工程设计、人工智能、多媒体、分布式等新领域,这些新领域需要有新的数据库技术来支持。因此各大数据库厂商都开始遵循标准 SQL 的数据库基础上添加了提高效率 and 可用性的新功能。如,Oracle 数据库引入了“并行”机制,开始向“关系-对象”型数据库变迁,从而形成新一代数据库系统的萌芽。目前,“关系-对象”型数据库正在发展中。

1.1.1 关系型数据库的相关概念

1. 表(关系)、记录、字段

所谓关系型数据库,是指采用关系模型来组织数据的数据库。关系模型是在 1970 年由 E. F. Codd 博士提出的,在之后的几十年中,关系模型的概念得到了充分的发展并逐渐成为数据库架构的主流模型。简单来说,关系模型指的就是二维表格模型,而一个关系型数据库就是由二维表及其之间的关系组成的一个数据组织。

在关系型数据库中,标准二维表是基本的数据存储单元,里边存储了实体的属性和实体间的关系。标准二维表在数据库中通常称为表(table),表中的任一行或一列都是单一的,不能被继续拆分,并且任意两行或两列都不能被合并。表中的每一行保存一个实体的所有属性值,称为一条记录。表中的每一列保存所有实体的同一个属性值,称为一个字段;该字段的首行是实体的属性名,称为字段名;该字段的其他所有行是属性值,称为字段值。如图 1-1 所示。

在图 1-1 的学生信息表中有 5 个字段(学生编号、姓名、性别、出生日期、所在班级),每个字段存储所有学生的某一方面属性值,如“姓名”字段存储了所有学生的姓名。该表中有四条记录,每条记录存储了某个学生的所有属性值,如第一条记录是学生李莉的所有信息。

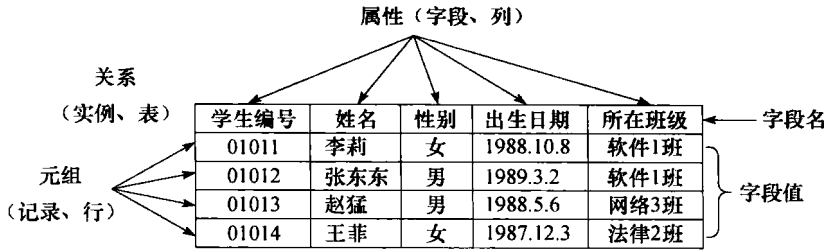


图 1-1 学生信息表

2. 主键

二维表中的每一行记录恰好存储了每个实体的属性，为了减少数据冗余，要求表中的记录不能重复存储，也就是说表中的任意两行记录中的数据不能完全相同，因为它们分别代表了不同的实体。为了达到这个目的，在表中一般都有一个叫做主键的字段，该字段的值在整个表中都不能重复，也就是表中任意两行的主键字段的值都不能相同，并且主键字段的值也不允许为空。这样，即使表中的其他所有字段值都相同，主键字段值也不同，因此可以保证表中的任意两行数据不完全相同。

在工作和生活中，我们用到的表通常会包含这样的代表不同数据的字段，如学生信息表的“学生编号”字段、课程信息表的“课程编号”字段、职工表的“职工编号”字段等。实际上，这样的字段并不是实体与生俱来的属性，而是为了管理需要而为实体增加的属性，其目的就是区分两个实体。这些字段就是表中的主键。每个表只能有一个主键，如图 1-2 所示。

在某些情况下，表中的每个字段都有可能重复或者必须重复，这样该表中的单个字段就不能区分任意两行记录，不能充当主键。在这种情况下，需要将多个字段组合在一起充当主键，即组合主键。在组合主键中，字段的组合值不能重复，组合中的任意一个字段值不能为空。如，学生成绩表有“学生编号”、“课程编号”和“成绩”三个字段。如果一个学生可以考多门课程，一门课程可以被多个学生考，那么表中的所有字段值都有可能重复，因此它们都不能单独充当主键。但是将“学生编号”和“课程编号”组合在一起就不会重复，也不应该重复，因此该表的主键就是“学生编号”和“课程编号”的组合主键，如图 1-3 所示。

主键

课程编号	课程名称	课时	学分
C1101	C++程序设计	110	6
C1102	数据结构	80	4
C1103	Oracle数据库	110	6

图 1-2 课程信息表

组合主键

学生编号	课程编号	成绩
01011	C1101	80
01011	C1102	90
01012	C1101	75
01012	C1102	87
01013	C1102	100

图 1-3 成绩表

3. 外键

在某些情况下，一个表中会包含另一个表的主键字段，作为本表中所描述实体的一个属性。如，学生表中包含系部信息表的系部编号、雇员表中包含部门表的部门编号、产品表中

包含种类表的种类编号等。这样的字段在本表中叫做外键，是联系另外一个表的纽带，可以通过该字段去另一个表中发现更多的信息。一个字段在表 A 中当做外键，在表 B 中当做主键或唯一键，那么相对来说，表 A 称做表 B 的子表，表 B 称做表 A 的父表。子表中的外键受父表中主键或唯一键的约束，如图 1-4 所示。

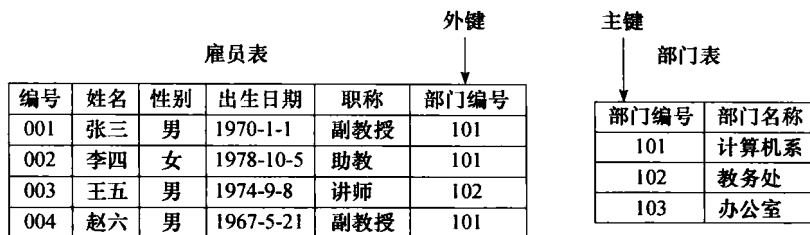


图 1-4 父表与子表

4. 表间关系

在关系型数据库中不仅可以存储数据，而且还可以存储两张表之间的关系（联系）。如果两张表具有相同的字段（字段值和含义相同，字段名不一定相同），那么这两张表之间就具有了关系。表和表之间有下列关系：

- 一对一关系（1:1） 有两个表，表 A 和表 B，表 A 中的一条记录在表 B 中有一条记录与之对应；反过来，表 B 中的一条记录在表 A 中也仅有一条记录与之对应，则称表 A 与表 B 之间有一对一关系。例如，对于雇员表和工资表，一个雇员只有一个工资，而一个工资只能属于一个雇员，则雇员表和工资表之间具有一对一关系。
- 一对多关系（1:n） 有两个表，表 A 和表 B，表 A 中的一条记录在表 B 中有多条记录与之对应；反过来，表 B 中的一条记录在表 A 中仅有一条记录与之对应，则称表 A 与表 B 之间有一对多关系。例如，对于部门表和雇员表，一个部门有多个雇员，而一个雇员只能属于一个部门，则部门表和雇员表之间具有一对多关系。
- 多对多关系（m:n） 有两个表，表 A 和表 B，表 A 中的一条记录在表 B 中有多条记录与之对应；反过来，表 B 中的一条记录在表 A 中也有多条记录与之对应，则称表 A 与表 B 之间有多对多关系。例如，对于学生表和课程表，一个学生可以选修多门课程，而一门课程可以被多个学生选修，则学生表和课程表之间具有多对多关系。

由于关系型数据库不能实现多对多的关系，所以在数据库设计时，必须通过增加第三张表将一个多对多关系转化为两个一对多关系。如前面提到的学生表和课程表之间需要增加学生课程成绩表，将原来的多对多关系转换为两个一对多关系，如图 1-5 所示。

5. 关系型数据库的完整性约束

关系完整性是为保证数据库中数据的正确性和相容性而对关系模型提出的某种约束条件或规则。完整性通常包括实体完整性、域完整性、参照完整性和用户自定义完整性，其中实体完整性、域完整性和参照完整性，是关系模型必须满足的完整性约束条件。

- 实体完整性 实体完整性是指表中的主键字段值不能重复也不能为空。表中的每一行描述了一个实体，因此表就是多个实体的集合。现实世界中的实体是可以相互区分、识别的，因为它们具有某种唯一性标识。在表中，也要区分任意两行记录，因

为它们分别代表了不同的实体。表中的主键字段作为唯一标识，不能重复并且不能为空，否则表明关系模式中存在着不可确定或不可标识的实体（因为空值是“不确定的”，这与现实世界的实际情况相矛盾，这样的实体就不是一个完整实体）。

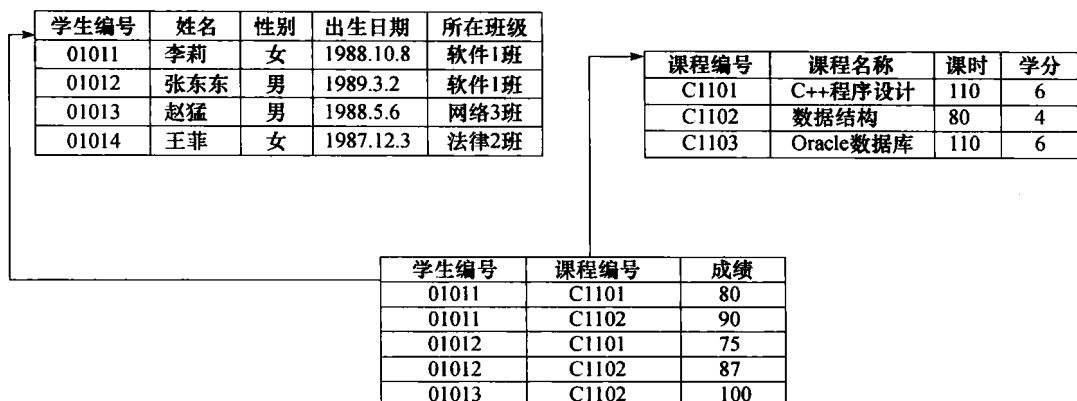


图 1-5 学生成绩关系表

- **域完整性** 域完整性是指字段（列）值域的完整性。如，从数据类型、格式、值域范围、是否允许空值等进行约束。域完整性限制了某些属性中出现的值，把属性限制在一个有限的集合中。
- **参照完整性** 参照完整性是指子表中的外键字段值受父表中主键字段或唯一键字段值的约束或限制，也就是表的外键约束。关系型数据库中通常包含具有关系的多个表，关系是通过两个表的相同字段（又称两表的公共字段）实现的，并且子表中公共字段的值应参照父表中公共字段的值。
- **用户自定义完整性** 实体完整性和参照完整性适用于任何关系型数据库系统，它们主要是针对关系的主键字段和外键字段的取值做出必须而有效的约束；而用户自定义完整性则是根据应用环境的要求和实际需要，对某一业务规则提出数据约束条件。用户自定义完整性主要是针对字段或记录的有效取值范围进行约束，如性别字段值只能是男或女，学生的成绩在 0 到 100 之间。

1.1.2 SQL 命令概述

SQL (Structured Query Language, 结构化查询语言) 是一种数据库查询和程序设计语言，用于查询、更新和管理关系型数据库。SQL 语言是高级的非过程化编程语言，允许用户在高层数据结构上工作。它不要求用户指定对数据的存放方法，也不需要用户了解具体的数据存放方式，所以具有完全不同底层结构的不同数据库系统，可以使用相同的 SQL 语言作为数据输入与管理的接口。

SQL 语言最早是 IBM 的圣约瑟研究实验室为其关系数据库管理系统 System R 开发的一种查询语言，它的前身是 SQUARE 语言。SQL 语言结构简洁、功能强大、简单易学，所以自从 IBM 公司 1981 年推出以来，SQL 语言就得到了广泛应用。美国国家标准局 (ANSI) 与国际标准化组织 (ISO) 已经制定了 SQL 标准。1992 年，ISO 和 IEC 发布了 SQL 国际标准，称为 SQL-92。ANSI 随之发布的相应标准是 ANSI SQL-92 (它有时被称为 ANSI SQL)。尽管不同的关系型数据库使用的 SQL 版本有一些差异，但大多数都遵循 ANSI SQL 标准。SQL 语

言包含 4 个部分：

- 数据定义语言 (Data Definition Language, DDL), 用于定义和管理数据库以及数据库中的各种对象, 例如数据库、数据表以及视图。DDL 语言通常包括对每个对象的 CREATE、ALTER、DROP 命令。
- 数据操作语言 (Data Manipulation Language, DML), 用于对数据库中存储的数据进行插入、修改和删除。DML 语言包括 INSERT、UPDATE、DELETE 命令。
- 数据查询语言 (Data Query Language, DQL), 用于对数据库中存储的数据执行各种查询操作, 包括 SELECT 命令。
- 数据控制语言 (Data Control Language, DCL), 用于在数据库中授予或回收某种访问权限, 并控制数据库操纵事务发生的时间及效果, 对数据库实行监视等。DCL 语言通常包括 GRANT、REVOKE、COMMIT、ROLLBACK 等语句。

1.2 数据库的设计

数据库设计的主要任务是通过现实世界中的数据进行抽象, 得到符合现实世界要求的、能被 DBMS 支持的数据库模型。在数据库设计阶段应该对需要存储的数据进行详细的调查、分析, 识别出真正需要存储的原始数据, 并根据这些数据设计出合理的表结构、表间关系以及其他数据库对象。

数据库设计的具体步骤如下：

- 1) 数据需求调查。根据系统的业务需求, 对用户需要存储和处理的数据进行详细调查。可以从用户日常的工作入手, 调查工作中填写的各种表格、编制的文档资料、上交的报表、传递的各类文件、保存的各类档案等。
- 2) 数据需求分析。根据数据需求调查阶段获取的资料进行分析, 找出用户真正需要存储和处理的原始数据。数据确定后, 可以请相关用户确认, 查找遗漏的、不准确的数据, 进一步完善数据资料。
- 3) 概念设计。通过数据抽象, 设计系统概念模型。将数据需求分析阶段获得的数据按照物理实体和属性进行分类、归纳, 绘制数据库的 E-R 模型来表达数据库的概念模型, 参见 1.2.1 节。
- 4) 逻辑结构设计。设计系统的模式和外模式, 对于关系模型主要是基本表和视图。可以利用前面设计的 E-R 模型导出表结构及其表间关系。并且根据系统的业务处理要求, 设计以表结构为基础的视图结构, 参见 1.2.2 节。
- 5) 优化数据库结构。利用数据库范式理论优化数据库结构, 降低数据冗余, 增强数据共享, 参见 1.2.3 节。
- 6) 物理结构设计。使用 DBMS 系统建立数据库、表、表间关系、视图、存储过程等多种数据库对象来满足系统的业务需求。

1.2.1 数据库设计的 E-R 模型

E-R (Entity-Relationship) 模型即实体-关系模型, 它是由 P. P. Chen 于 1976 年首先提出的。它提供不受任何 DBMS (Database Management System) 约束的面向用户的表达方法, 在数据库设计中被广泛用作数据建模的工具。E-R 模型用来描述实体的属性和实体之间的关系, 主要包括三种模型元素: 实体、属性和关系。