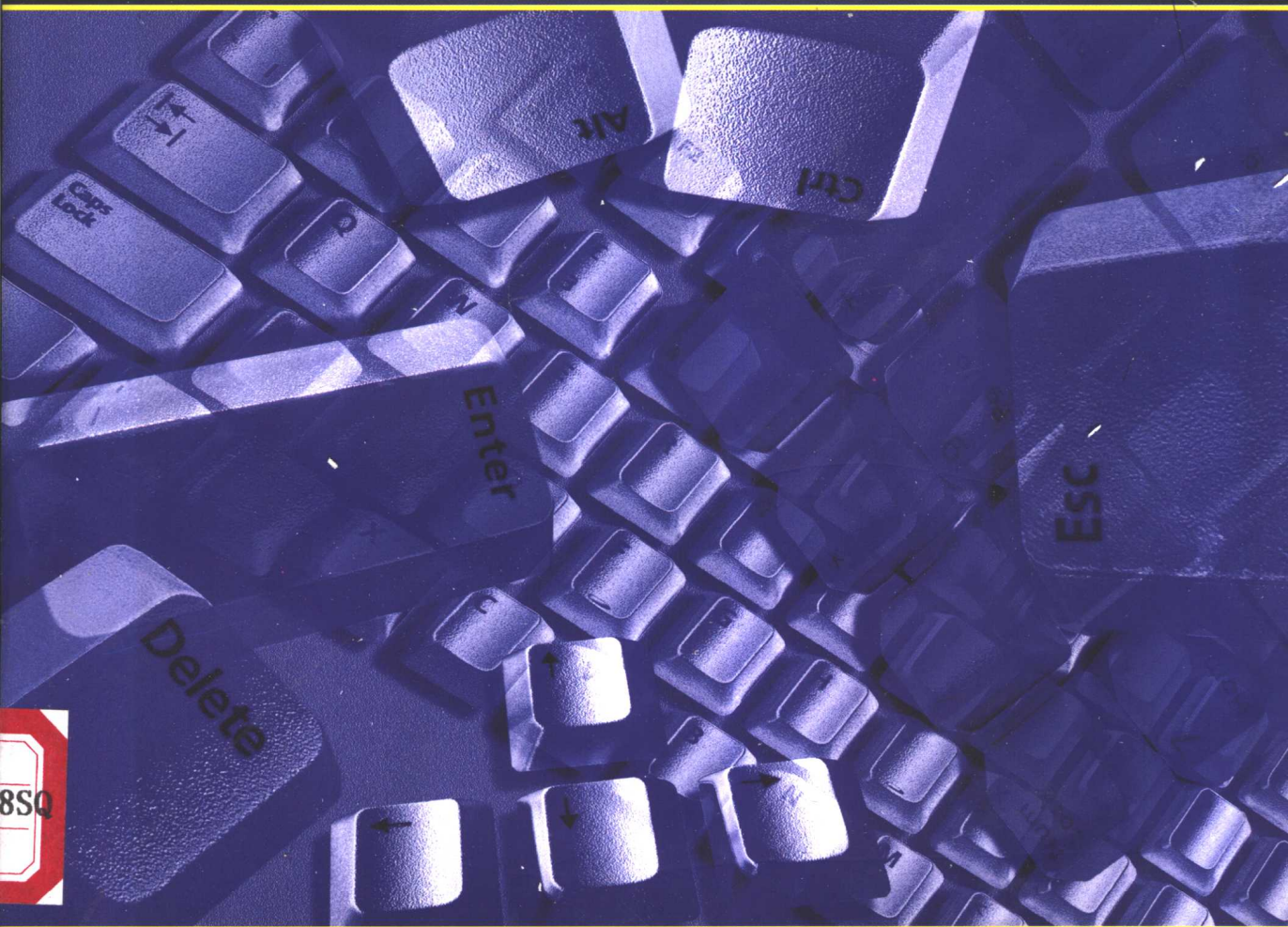


Oracle9i OCP DBA 考试指南丛书之一

# SQL 与 PL/SQL

## 程序设计基础

■ 冯凤娟 编著



清华大学出版社

Oracle9i OCP DBA 考试指南丛书之一

# SQL 与 PL/SQL 程序设计基础

冯凤娟 编著

清华大学出版社

(京)新登字 158 号

## 内 容 简 介

本书全面系统地介绍了 Oracle9i 数据库中使用的 SQL 命令 DDL、DML、DCL 语句的语法和功能, 以及如何用 DDL 语句管理数据库对象, 如何用 DML 语句操纵和查询数据库对象。同时, 介绍了 PL/SQL 的条件语句、循环语句、出错处理等语句的语法和使用规则, 以及如何用 PL/SQL 语句编写存储过程、函数、包、数据库触发器, 提高运行效率。

本书是由具有 8 年 Oracle 培训经验的教师根据多年的教学经验和 OCP 考试的风格特点编写而成的。本书不仅是参加 OCP 考试人员的首选参考书, 而且也可以作为从事 Oracle 数据库应用开发人员、管理人员、Oracle 初学者和大专院校师生的参考资料。

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: SQL 与 PL/SQL 程序设计基础

作 者: 冯凤娟 编著

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: 北京通州大中印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 22 字数: 504 千字

版 次: 2002 年 11 月第 1 版 2002 年 11 月第 1 次印刷

书 号: ISBN 7-302-05836-9/TP·3455

印 数: 0001~5000

定 价: 26.00 元

# Oracle9i OCP DBA 考试指南丛书序

随着信息技术的迅猛发展，越来越多的企业注重自身的信息化建设，数据库系统是企业信息资讯的核心和仓库。而 Oracle 是数据库领域的一支生力军，越来越多的企业都使用 Oracle 作为企业信息的核心和仓库。这就要求有大量熟悉 Oracle 系统管理和应用开发的专家来管理和开发企业的应用系统。OCP 就是为获得系统管理专家（DBA）和系统开发专家（DBO）全球认证资格的考试。

我们编写此系列丛书就是以大家比较关心的 OCP DBA 考试（即全球的 Oracle 认证考试）的五门课（SQL 与 PL/SQL 程序设计基础、Oracle 数据库体系结构和管理、Oracle 数据库备份和恢复、Oracle 数据库性能调整、Oracle 数据库网络管理）为主线，用图文并茂的方式、简捷明快的手法和通俗易懂的语言系统地介绍了 SQL 基础知识、PL/SQL 编写过程、函数、包、触发器、Oracle 体系结构、存储分配原则、安全管理、备份与恢复的各种方法、性能调整的几个方面、网络连接和配置复制等。本丛书共 5 本，分别对应 Oracle OCP DBA 的 5 门考试科目，适合于不同层次的读者及同一读者不同阶段学习的需要。

《SQL 与 PL/SQL 程序设计基础》主要针对 Oracle 的开发和系统管理人员，系统地讲述了 SQL 语言中的 DDL 语句、DML 语句的语法，介绍如何用 PL/SQL 将 SQL 语句组织成一个程序块，完成循环、条件判断、出错处理等功能，介绍如何用 PL/SQL 编写存储过程、函数、包、数据库触发器，以及如何调用存储在数据库中的过程、函数、包，如何触发触发器。

《Oracle 数据库体系结构和管理》主要针对 Oracle 系统管理人员，从不同的角度讲述 Oracle 数据库的组成（包括物理结构、内存结构、进程结构、逻辑结构等）和它们的管理，介绍 Oracle 数据库的启动和关闭的步骤和原理、数据库的存储管理、数据库的安装步骤和创建方法，以及数据库的安全管理（包括用户管理、权限管理、资源管理等）等。

《Oracle 数据库备份和恢复》主要针对 Oracle 系统管理人员，介绍 Oracle 数据库的各种备份方法（包括物理备份和逻辑备份的各种方法以及它们的适用范围）和各种恢复方法（包括物理恢复和逻辑恢复的各种方法以及它们的适用范围），介绍如何用恢复管理器（RMAN）进行各种恢复操作。

《Oracle 数据库性能调整》主要针对 Oracle 系统管理人员，介绍影响数据库性能的各种因素以及各种因素的重要性排列，介绍如何动态查询系统的状态，以及如何从操作系统级、数据库核心级、数据库设计级、SQL 语句级等多个方面对这些因素进行调整以改善数据库的性能。

《Oracle 数据库网络管理》主要针对 Oracle 系统管理人员和开发人员，介绍为了客户/服务器之间的通信，如何配置客户和服务器两端的配置文件，客户机如何连接到服务器，如何建立数据库链路，如何利用数据库链路实现多台服务器之间数据的复制，Oracle 命名

服务器的使用和配置，多线程服务器的使用和配置，以及网络的安全性问题等。

本书具有以下特点：

**内容新颖全面：**本书以最新的 Oracle9i 为叙述对象，系统而全面地介绍了 Oracle 数据库的分析、设计、开发、管理等方面的内容。

**手法简洁明快：**本书以 OCP 考试的五门课程为纵线，再横线展开对每门课程进行详细地阐述，脉络清晰，层次分明，写作手法相当简洁。

**语言通俗易懂：**本书由具有多年 Oracle 培训和开发经验的教师编写而成，其丰富的经验充分体现在语言的运用上，通俗易懂、形象生动的语言立刻使深奥的理论变得简单明了。

**实例典型实用：**书中所举实例，均具代表性、典型性，读者通过实例可加深对相应技术点的理解，做到融会贯通。

相信通过本套丛书的学习，您一定能对 Oracle 数据库有一个全面的、深入的理解，能够利用所学知识顺利地通过 OCP 考试，获得 Oracle 全球认证的数据库管理员专家（DBA）证书，成为一位优秀的 Oracle 数据库专家。

最后真诚地感谢参与本书编写的各位作者，感谢在本丛书的编写过程中给予关心和支持的同仁，感谢清华大学出版社的编辑，是他们的共同努力才使本书得以出版。

编 者

2002 年 6 月

# 前 言

随着信息技术的迅猛发展，越来越多的企业注重自身的信息化建设，数据库系统是企业信息资讯的核心和仓库。而 Oracle 是数据库领域的一支生力军，越来越多的企业都使用 Oracle 作为企业信息的核心和仓库。这就要求有大量熟悉 Oracle 系统管理和应用开发的专家来管理和开发企业的应用系统。OCP 就是为获得系统管理专家(DBA)和系统开发专家(DBO)全球认证资格的考试。

深入了解 Oracle9i 数据库中使用的 SQL 和 PL/SQL 命令的语法、功能和使用规则，来管理和操纵数据库对象，以及如何编写存储过程、函数、包、数据库触发器，来提高运行效率和增强系统完整性，对于一个优秀的数据库管理人员和开发人员来说是非常必要的，也是参加 OCP DBA 考试的五门课程之一。

本书共分 8 章，系统地介绍了数据库设计模型、Oracle9i 数据库中使用的 SQL 命令 DDL、DML、DCL 语句的语法和功能，如何用 DDL 语句管理数据库对象，如何用 DML 语句操纵和查询数据库对象，也介绍了 PL/SQL 的条件语句、循环语句、出错处理等各种语句的语法和使用规则，以及面向对象的 PL/SQL 语句的语法和实例。最后介绍了如何用 PL/SQL 语句编写存储过程、函数、包、数据库触发器，提高运行效率和增强系统的完整性。

通过本书的学习，可以帮助您顺利地通过 OCP 考试，获得 Oracle 全球认证的数据库管理员专家(DBA)证书，从而成为一位优秀的 Oracle 数据库专家。

本书由冯凤娟老师编写，由于时间仓促和水平有限，书中如有疏漏之处，恳请广大读者批评指正。

编 者  
2002 年 6 月

# 目 录

|              |                                   |           |
|--------------|-----------------------------------|-----------|
| <b>第 1 章</b> | <b>数据库设计概述与 SQL*Plus 基础</b> ..... | <b>1</b>  |
| 1.1          | 数据库系统发展的历程和开发过程.....              | 1         |
| 1.1.1        | 计算机数据管理的几个演变阶段.....               | 1         |
| 1.1.2        | 数据库系统发展的历程.....                   | 1         |
| 1.1.3        | 应用系统的开发周期.....                    | 2         |
| 1.2          | 数据模型和数据库设计.....                   | 3         |
| 1.2.1        | 逻辑模型.....                         | 3         |
| 1.2.2        | 物理设计.....                         | 5         |
| 1.3          | SQL、SQL*Plus 与 PL/SQL 命令.....     | 8         |
| 1.3.1        | SQL 命令.....                       | 8         |
| 1.3.2        | SQL*Plus 命令.....                  | 9         |
| 1.3.3        | PL/SQL 命令.....                    | 17        |
| 1.4          | 习题.....                           | 18        |
| <b>第 2 章</b> | <b>查询</b> .....                   | <b>19</b> |
| 2.1          | 简单查询.....                         | 19        |
| 2.1.1        | 运算符与表达式.....                      | 20        |
| 2.1.2        | SELECT 子句.....                    | 24        |
| 2.1.3        | FROM 子句.....                      | 27        |
| 2.1.4        | WHERE 子句.....                     | 28        |
| 2.1.5        | ORDER BY 子句.....                  | 33        |
| 2.1.6        | SELECT 语句中的表达式.....               | 34        |
| 2.1.7        | GROUP BY 和 HAVING 子句.....         | 35        |
| 2.1.8        | START WITH 和 CONNECT BY 子句.....   | 37        |
| 2.2          | 高级查询.....                         | 39        |
| 2.2.1        | 多表连接.....                         | 39        |
| 2.2.2        | 集合运算.....                         | 44        |
| 2.2.3        | 子查询.....                          | 49        |
| 2.3          | 函数.....                           | 57        |
| 2.3.1        | 单行函数.....                         | 58        |
| 2.3.2        | 聚组函数.....                         | 85        |
| 2.3.3        | 函数的嵌套.....                        | 88        |

|              |                       |            |
|--------------|-----------------------|------------|
| 2.4          | 查询结果的报表输出 .....       | 89         |
| 2.4.1        | 格式化列 .....            | 90         |
| 2.4.2        | 用空行和总计使报表简洁 .....     | 91         |
| 2.4.3        | 定义页标题和页面的大小 .....     | 92         |
| 2.5          | 习题 .....              | 94         |
| <b>第 3 章</b> | <b>数据库对象的管理 .....</b> | <b>103</b> |
| 3.1          | 表 .....               | 103        |
| 3.1.1        | 创建基表 .....            | 103        |
| 3.1.2        | 修改基表 .....            | 123        |
| 3.1.3        | 删除表、重新命名表 .....       | 129        |
| 3.2          | 视图 .....              | 131        |
| 3.2.1        | 创建视图 .....            | 132        |
| 3.2.2        | 删除视图 .....            | 136        |
| 3.2.3        | 对视图的操作 .....          | 137        |
| 3.2.4        | 确认视图的名字和结构 .....      | 138        |
| 3.3          | 索引 .....              | 138        |
| 3.3.1        | B*树索引 .....           | 139        |
| 3.3.2        | 位图索引 .....            | 142        |
| 3.4          | 聚簇 .....              | 143        |
| 3.4.1        | 创建聚簇 .....            | 144        |
| 3.4.2        | 删除聚簇 .....            | 144        |
| 3.5          | 序列生成器 .....           | 145        |
| 3.5.1        | 建立序列生成器 .....         | 145        |
| 3.5.2        | 使用序列生成器 .....         | 146        |
| 3.5.3        | 修改序列生成器 .....         | 147        |
| 3.5.4        | 删除序列生成器 .....         | 148        |
| 3.5.5        | 确认序列生成器 .....         | 148        |
| 3.6          | 同义词 .....             | 149        |
| 3.6.1        | 同义词的概念 .....          | 149        |
| 3.6.2        | 创建同义词 .....           | 149        |
| 3.6.3        | 删除同义词 .....           | 150        |
| 3.6.4        | 确认同义词的信息 .....        | 150        |
| 3.7          | 习题 .....              | 150        |
| <b>第 4 章</b> | <b>操纵数据库对象 .....</b>  | <b>156</b> |
| 4.1          | 插入数据 .....            | 156        |
| 4.2          | 更新数据 .....            | 159        |



---

|              |                                 |            |
|--------------|---------------------------------|------------|
| 4.3          | 删除数据 .....                      | 161        |
| 4.3.1        | 删除关系表的数据 .....                  | 161        |
| 4.3.2        | TRUNCATE 一张表 .....              | 162        |
| 4.4          | 锁定表 .....                       | 163        |
| 4.4.1        | selecting rows FOR UPDATE ..... | 163        |
| 4.4.2        | LOCK A TABLE .....              | 163        |
| 4.5          | 事务控制 .....                      | 163        |
| 4.5.1        | 提交(COMMIT) .....                | 165        |
| 4.5.2        | 回退(ROLLBACK) .....              | 166        |
| 4.5.3        | 保存点(SAVEPOINT) .....            | 166        |
| 4.5.4        | 语句级回退 .....                     | 167        |
| 4.6          | 运行时指定变量 .....                   | 167        |
| 4.6.1        | 使用&置换变量 .....                   | 167        |
| 4.6.2        | 定义用户变量 .....                    | 169        |
| 4.7          | 习题 .....                        | 172        |
| <b>第 5 章</b> | <b>面向对象的 PL/SQL 设计 .....</b>    | <b>174</b> |
| 5.1          | 操纵对象表和含有对象类型的关系表 .....          | 174        |
| 5.1.1        | 创建对象类型 .....                    | 174        |
| 5.1.2        | 数据库中更改和删除对象 .....               | 176        |
| 5.1.3        | 调用数据库中对象的方法 .....               | 177        |
| 5.1.4        | 对象类型使用%TYPE .....               | 177        |
| 5.1.5        | 在 DML 语句中使用对象 .....             | 178        |
| 5.1.6        | 在 PL/SQL 中使用对象 .....            | 181        |
| 5.2          | 嵌套表 .....                       | 184        |
| 5.2.1        | 创建嵌套表 .....                     | 184        |
| 5.2.2        | 操纵整张表 .....                     | 186        |
| 5.2.3        | 操纵存储表 .....                     | 188        |
| 5.3          | 变长数组(VARRAY) .....              | 190        |
| 5.3.1        | 创建变长数组 .....                    | 190        |
| 5.3.2        | 操纵变长数组 .....                    | 191        |
| 5.4          | 索引表 .....                       | 191        |
| 5.5          | VARRAY、嵌套表和索引表之间的比较 .....       | 192        |
| 5.6          | 集合方法 .....                      | 193        |
| 5.7          | 习题 .....                        | 199        |
| <b>第 6 章</b> | <b>PL/SQL 基础 .....</b>          | <b>203</b> |
| 6.1          | PL/SQL 的简介 .....                | 203        |

|              |                 |            |
|--------------|-----------------|------------|
| 6.1.1        | PL/SQL 的优点      | 204        |
| 6.1.2        | PL/SQL 的特点      | 205        |
| 6.2          | PL/SQL 的程序结构    | 206        |
| 6.2.1        | 块结构             | 206        |
| 6.2.2        | 变量定义            | 208        |
| 6.2.3        | 编程的指导原则         | 218        |
| 6.2.4        | PL/SQL 中的运算符和函数 | 220        |
| 6.3          | PL/SQL 的控制结构    | 224        |
| 6.3.1        | 条件语句            | 225        |
| 6.3.2        | 循环              | 230        |
| 6.3.3        | 标签(LABEL)       | 233        |
| 6.4          | 出错处理            | 237        |
| 6.4.1        | 触发和处理错误         | 237        |
| 6.4.2        | 各种错误处理          | 241        |
| 6.5          | 游标              | 247        |
| 6.5.1        | 显式游标            | 248        |
| 6.5.2        | 隐式游标            | 258        |
| 6.5.3        | 显式游标与隐式游标的比较    | 259        |
| 6.6          | PL/SQL 与数据库的交互  | 260        |
| 6.6.1        | 查询命令            | 260        |
| 6.6.2        | 操纵数据命令          | 261        |
| 6.6.3        | DML 语句的结果       | 262        |
| 6.6.4        | 事务控制命令          | 263        |
| 6.7          | 习题              | 264        |
| <b>第 7 章</b> | <b>过程、函数、包</b>  | <b>279</b> |
| 7.1          | 存储过程、函数         | 279        |
| 7.1.1        | 存储子程序与应用子程序的区别  | 279        |
| 7.1.2        | 开发一个存储过程和函数     | 280        |
| 7.1.3        | 过程和函数中的出错处理     | 286        |
| 7.1.4        | 过程和函数的管理        | 289        |
| 7.1.5        | 参数传值            | 292        |
| 7.1.6        | 过程和函数的调用        | 293        |
| 7.1.7        | 过程和函数的测试        | 296        |
| 7.1.8        | 过程和函数的安全性       | 298        |
| 7.1.9        | 过程和函数的依赖性       | 299        |
| 7.1.10       | 过程和函数的优点        | 300        |
| 7.2          | 包               | 301        |

---

|              |                             |            |
|--------------|-----------------------------|------------|
| 7.2.1        | 包的组成 .....                  | 301        |
| 7.2.2        | 开发包 .....                   | 302        |
| 7.2.3        | 包的管理 .....                  | 305        |
| 7.2.4        | 包的调用 .....                  | 306        |
| 7.2.5        | 包的依赖性 .....                 | 307        |
| 7.2.6        | 包的优点 .....                  | 308        |
| 7.2.7        | 系统包 .....                   | 308        |
| 7.3          | 习题 .....                    | 309        |
| <b>第 8 章</b> | <b>触发器 .....</b>            | <b>311</b> |
| 8.1          | 触发器的基本概念和组成 .....           | 311        |
| 8.2          | DML 触发器 .....               | 313        |
| 8.2.1        | DML 触发器的组成和类型 .....         | 313        |
| 8.2.2        | 创建 DML 触发器 .....            | 315        |
| 8.2.3        | DML 触发器的管理 .....            | 320        |
| 8.2.4        | 测试 DML 触发器 .....            | 322        |
| 8.2.5        | 在 DML 触发器中进行的数据操作 .....     | 322        |
| 8.2.6        | DML 触发器的应用 .....            | 326        |
| 8.3          | 数据库事件触发器 .....              | 330        |
| 8.3.1        | 定义数据库事件或模式事件触发器的语法 .....    | 331        |
| 8.3.2        | 事件的属性 .....                 | 331        |
| 8.3.3        | 数据库事件触发器 .....              | 332        |
| 8.3.4        | DDL 事件触发器 .....             | 333        |
| 8.3.5        | INSTEAD OF 触发器(替代触发器) ..... | 334        |
| 8.4          | 习题 .....                    | 336        |
| <b>附录</b>    | <b>习题答案 .....</b>           | <b>337</b> |

# 第 1 章 数据库设计概述与 SQL\*Plus 基础

## 1.1 数据库系统发展的历程和开发过程

Oracle9i 是一个对象关系型数据库管理系统(ORDBMS),它提供了解决关键信息管理问题的工具。在此将介绍计算机数据管理的几个演变阶段,以及数据库系统发展的历程。

### 1.1.1 计算机数据管理的几个演变阶段

计算机数据管理最早采用的是人工管理阶段,在这一阶段中没有支持数据管理的软件,在程序中要规定数据的逻辑结构和物理结构,数据与程序不独立。数据组织面向应用,数据不能共享导致数据重复。数据处理的方式是批处理。

后来发展到文件系统阶段,在这一阶段中信息以文件为单位存储在外存,且由 OS 统一管理。逻辑结构与物理结构脱钩,使数据与程序有了一定的独立性。实现了以文件为单位的共享。数据组织(逻辑结构)仍面向应用,所以不同应用的数据存在冗余。文件之间相互独立,不能反映现实世界中事物之间的联系。数据处理的方式可以分时处理或批处理。

最后发展到数据库系统阶段,在这一阶段中面向企业或部门以数据为中心组织数据,形成综合性的数据库,为各应用共享。数据库冗余小,节省存储空间,有利于保持数据的一致性。程序和数据具有较高的独立性,数据易修改、易扩充,具有良好的用户接口,用户可方便地开发和使用数据库。提供了数据的安全性、完整性以及并发控制。

### 1.1.2 数据库系统发展的历程

什么是数据库?数据库是数据集中存放的地方,并且只有经过授权的人或程序才能使用数据库中的信息,才能对数据库进行操作,在数据库中的数据经过适当设计,可以减少信息冗余。每一个数据库系统都有自己的数据库管理系统对数据进行管理。数据库管理系统(DBMS)是一个软件系统,它有以下功能——存储、检索和修改数据,保证数据的一致性和完整性,解决并发问题,支持对数据操作的各种接口,进行数据管理等。

数据库系统的发展分四个阶段:(1)在 20 世纪 60 年代,采用的是层次型数据库系统,它支持一父多子关系(one-parent-multiple-children relationships)。(2)到 70 年代,采用的是网状型数据库系统,它支持多父多子关系(multiple-parents-multiple-children-relationships)。(3)到 80 年代,采用的是关系型数据库系统,起源于 IBM 的 ALPHA 模型(E.F.Codd, 1970 提出的),系统自动建立起所有可能的关系,1979 年推出的 Oracle V2 是最早的商业关系型

数据库系统。(4)到 90 年代末期,提出了对象关系型数据库系统,允许定义数据方法和数据类型,通过方法对对象进行操作。由于关系型数据库系统的使用非常简单并且结构容易修改,所以现在用的最多的是关系型数据库系统和对象关系型数据库系统。

关系型数据库系统的特点是:(1)结构化。在数据库中数据是以表(Table)的形式存储和访问的,表与表之间的主外键关系体现了数据之间的关系。(2)可操作性。支持第四代语言(4GL),即非过程化的类似英语的语言(例如:SQL),用 SQL 可以很方便地检索和修改数据,改变数据库的结构,控制对数据的访问。(3)数据完整性规则。在表上可以定义各种完整性规则保证数据的一致性和正确性,其中表上的主键和外键就是完整性规则的例子。(4)拥有集成的数据字典,管理数据的存储和用户的操作,支持所有关系运算。

关系对象型的数据库系统的特点是:(1)依然对关系型数据库提供支持,即关系型数据库中的所有操作完全可以执行。(2)支持非结构化数据,利用大对象类型(LOB)存储非结构化数据,并且允许数据达到 4GB。(3)允许用户自定义数据类型,如对象类型、引用类型、变长数组、嵌套表等类型。(4)可以通过方法对对象进行操作,为开发人员提供更多的灵活性和控制手段。

### 1.1.3 应用系统的开发周期

一个好的数据库设计可以使得应用开发和性能调整很容易实现。在进入数据库详细设计之前,先来回顾一下应用开发的几个阶段。每一个应用系统,不管大小,都要经过系统开发周期的下列几个阶段。

(1) 分析阶段:分析是系统开发的第一阶段。在这一阶段需要回答以下问题:为什么要开发系统?谁会使用此系统?开发系统对用户有什么好处?业务规则和需求是什么?在需求分析阶段需要决定系统完整的功能,构造系统的逻辑模型。一般情况下,功能层的管理员(业务员)参与这一阶段。本阶段产生需求分析报告。

(2) 设计阶段:设计阶段是系统开发的最重要的阶段。在系统的需求分析以后,就开始了设计阶段。在这一阶段,使用 ER 图根据模型设计数据库,也就是将逻辑数据库设计转换成物理结构。关于 ER 图和逻辑设计将在下一节介绍。本阶段产生设计文档。

(3) 开发阶段:设计阶段就是根据上一阶段的设计进行编程和实施;同时编写用户手册、帮助屏幕、操作手册等文档。

(4) 测试阶段:测试就是确定开发的应用能否完成设计时的要求。需要对整个系统做系统测试和完整性测试以及错误的修改,测试工作需要重复多次。测试阶段需要系统的最终用户参与,同时产生测试报告。

(5) 实施阶段:实施阶段是开发周期的最后阶段,一旦测试完成了,系统就可以开始实施了。实施后产生的错误报告反馈给相应的阶段。所有这些步骤都可以重复执行以完善系统。

图 1-1 显示了应用系统开发生命周期。

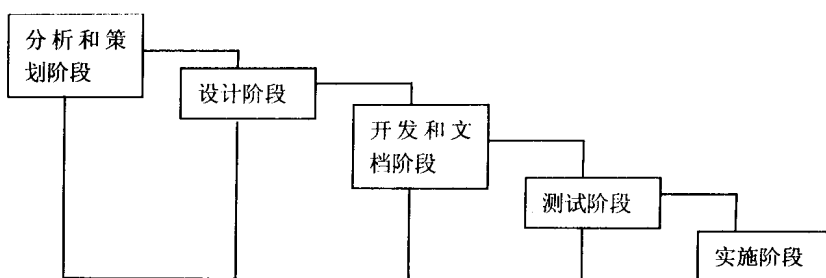


图 1-1 应用系统开发生命周期

## 1.2 数据模型和数据库设计

在建表之前，需要设计数据库结构，需要了解数据处理的模型，并且将实体关系的逻辑模型转换成关系数据库的物理模型。

### 1.2.1 逻辑模型

在系统开发周期的设计阶段建立数据库的逻辑模型。一个 RDBMS 的逻辑模型一般就是指实体和关系图，也叫 ER 图。一个 ER 模型显示了要建模的数据内部及数据之间的相互关系、特性、结构。建立实体关系(ER)模型的优点是：

- (1) 便于在不同的人之间交流自己的观点。
- (2) 可以将一个单位的信息需求进行采集，并且提供精确的资料。
- (3) 为系统提供一个很容易理解的图形化的布局图。
- (4) 很容易被开发和加工。
- (5) 很清楚地定义了信息需求的范围。
- (6) 将业务所需的信息与业务的活动进行了分离，为多个应用、开发工程以及购买的应用程序的集成提供了一个有效的平台。

#### 1.2.1.1 ER 模型的组成

一个 ER 模型由三部分组成。

- (1) 实体：是一个有意义的人、地方、事物。例如：客户、销售代表、订单。
- (2) 关系：两个实体之间的关系。例如：订单与购买物品、客户与销售代表之间的关系。
- (3) 属性：描述一个实体的特性。例如：客户的姓名、电话号码、身份证号码等。

这三者之间的关系可以用一个实例加以说明。假如有一个公司 ABC，在美国有许多办事处(OFFICE)，在每一个办事处有许多部门(DEPARTMENT)，每一个部门有许多雇员(EMPLOYEE)。用 ER 模型的术语来理解，可以用三个实体来表示：OFFICE、DEPARTMENT、EMPLOYEE。每一个实体都有自己的特性，例如，当说到办事处时，可能需要知道办事处

所在的城市(CITY)和洲(STATE), 以及在该办事处工作的雇员数等属性。类似地, 当了解部门信息时可能需要了解部门的名称、部门的经理等属性。当了解雇员信息时可能需要了解雇员的名称、出身日期、所在的部门等属性。

在每一个实体上定义一个唯一地表示了该实体的唯一标识符 UID(UNIQUE IDENTIFIER)。UID 是属性之间的组合, 或者关系之间的组合, 或者属性和关系之间的组合。例如, 有两个相同名称的雇员, 如何区分他们? 对办事处来说, 城市和洲可以唯一地表示每一个办事处; 对部门来说, 部门名称唯一地表示了部门; 对雇员来说, 引入雇员编号唯一地表示了雇员信息。

在这些实体的属性中有些是必须有的, 有些是可选的, 有些是唯一地表示了该实体的。其中\*号代表了必须有的属性, #号代表了唯一的属性, 小写字母 o 代表了可选的属性。

假设在某一个城市有一个办事处, 该办公室至少有一个部门。在一个部门中, 有可能一个雇员都没有。在 ER 模型图中, 用实线表示必须有的关系, 虚线表示可选的关系, 三角表示了一对多的关系。

如图 1-2 所示显示了这三个实体之间关系的 ER 图。

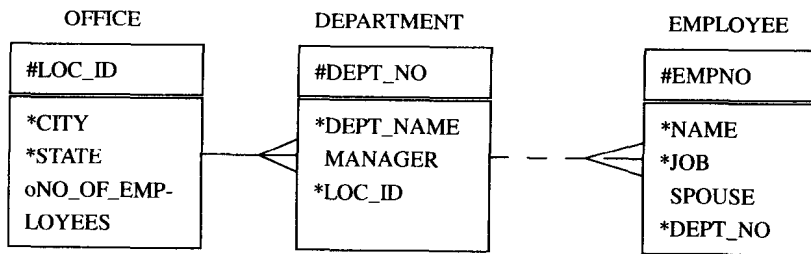


图 1-2 三个实体之间关系的 ER 图

### 1.2.1.2 实体之间关系的类型

两个实体之间的关系有以下三种类型。

(1) 一对一: 一对一的关系代表了一个实体中的每一种情况只表示了另一个实体的一种情况。例如, 每一个人与他的社会保险号码是一对一的关系, 一个人只能有一个社会保险号码, 一个社会保险号码只属于一个人。

(2) 一对多: 一对多的关系代表了一个实体中的每一种情况在另一个实体中表示了多种情况。例如, 部门和雇员的关系是一对多的关系, 一个部门可以有多个雇员, 而一个雇员只能属于一个部门。

(3) 多对多: 多对多的关系代表了一个实体中的每一种情况在另一个实体中表示了多种情况, 在第二个实体中的每一种情况也可能在第一个实体中表示了多种情况。例如, 医生和病人之间的关系是多对多的关系, 一个病人有很多医生, 一个医生有很多病人。



**注意:** 多对多的关系在 RDBMS 中是不存在的, 因为无法正确地表示这种连接。为了解决这种问题, 需要建立另一个实体, 该实体与第一个实体有一对多的关系, 与第二个实体有一对多的关系。

图 1-3 显示了这三种关系类型。

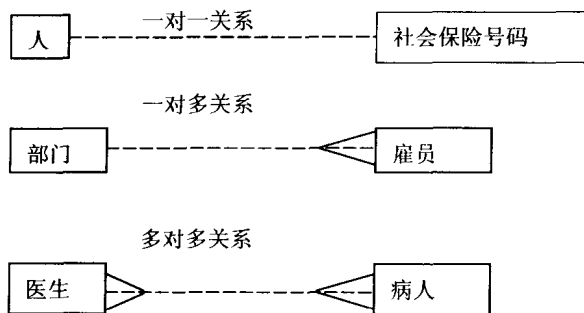


图 1-3 三种关系类型

### 1.2.1.3 规范化设计

当建立好了实体之间的关系图后，下一步工作是将设计规范化。规范化实际上就是从实体中删除冗余的信息，修改数据模型以便支持不同的功能需求，直到可以惟一地表示实体中的每一种情况为止。但是由于性能和实施问题，有时可能无法实现。在这种情况下，有可能仍然保留些冗余的信息。

规范化的优点是：数据的冗余最小化，减少由于没有完整性约束所产生的问题，识别缺少的实体、关系和表。

规范化的规则有三个。

一范式(1NF)：实体的所有属性必须是单值的并且不允许重复。

二范式(2NF)：实体的所有属性必须依赖整个实体的惟一标识。

三范式(3NF)：非惟一标识属性不允许依赖另一个非惟一标识属性。

一般都采用三范式进行数据库的设计，以便保证数据的冗余最小化，提高数据的完整性。

### 1.2.2 物理设计

使用逻辑模型建立物理模型就是建立数据库和数据库对象来表示实体和它们之间的关系。在物理模型中，每一个实体成为一张表，实体的属性成为表中的列，实体之间的关系成为表的一个或多个约束条件。物理实施可能要求对实体进行组合、分离，或者建立完全新的实体以便更好地实现逻辑模型。一个实体的惟一标识成为表的主键(PRIMARY KEY)，也可以建立存储过程、函数、触发器保证业务规则。

数据库设计阶段就是为一个关系数据库产生设计说明，包括表、索引、视图以及存储空间的定义。

将实体关系模型映射成一张表的步骤如下：

(1) 将实体映射成一张表。

(2) 将属性映射成列并且设置演示数据，明确地标识列名和相应的数据类型，如字符、数字、日期等。



(3) 将惟一标识符映射成一个主键。确定引用此主键的外键。

(4) 将关系映射成外键(FOREIGN KEY)。

一张表由一些行和列组成, 可以通过执行 SQL 命令操纵每一行的数据。一张表中的每一行由主键(PK)惟一地标识。可以通过外键(FK)将多个表的信息在逻辑上建立连接。建立主外键关系时要遵循下列原则:

(1) 主键的值不允许重复。

(2) 主键值一般不能修改。

(3) 外键值依赖于主键值, 纯粹是逻辑上的依赖, 没有物理指针。

(4) 一个外键值必须与已经存在的主键值匹配, 否则是 NULL。

至此, 我们已经引入了数据库设计时要涉及到的一些对象和相应的术语。数据库中常用的一些术语如表 1-1 所示。

表 1-1 数据库术语

| 术 语             | 描 述   |
|-----------------|---|
| TABLE           | 表是 RDBMS 的基本存储结构, 由一列或多列以及一行或多行组成                 |
| ROW             | 一行是由表中一组列值组成, 一行有时候叫做记录                           |
| COLUMN          | 一列代表了一张表中的一类数据。描述了列名和该列的数据类型和长度                   |
| FIELD           | 一行和一列交叉的地方就是域。域可以包含数据, 也可以没有数据, 没有数据就认为包含空值(NULL) |
| PRIMARY KEY(PK) | 一个主键是惟一标识表中每一行的一个列或一组列。例如: DEPTNO 等, 主键必须包含值      |
| FOREIGN KEY(FK) | 一个外键是参考同一张表或其他表中的主键的一个列或一组列。建立外键保证关系数据库建立时的一些业务规则 |

在将实体关系模型映射成一张表时应遵循以下指导方针:

(1) 表名应该很容易追溯到实体名。

(2) 列名也应该很容易追溯到实体关系模型。较短的列名可以减少分析 SQL 命令所用的时间。

(3) 应该规定自己的命名习惯和标准。不用 SQL 保留字, 如 NUMBER 等。一般用于记录表上列的约束条件和主外键时的符号如表 1-2 所示。

表 1-2 表上列的约束条件和主外键符号

| 符 号      | 含 义          |
|----------|--------------|
| PK       | 主键列          |
| FK       | 外键列          |
| FK1, FK2 | 同一个表中的两个外键   |
| FK1, FK1 | 同一个组合外键中的两个列 |
| NN       | 非空列          |
| U        | 惟一标识列        |
| U1, U1   | 组合惟一标识中的两列   |