

# 数据库课程设计与开发实操

刘福江 张传维 等 编著



科学出版社

# 数据库课程设计与开发实操

刘福江 张传维 等 编著

科学出版社

北 京

## 内 容 简 介

本书主要以初学者为主要读者对象,注重实际操作部分,对理论知识进行精简介绍。在上机部分,主要注重数据库基础知识的实际操作,以便进行后面数据库实际案例的设计。本书的精华部分就是将 Java、PHP 语言和 MySQL、SQL Server 数据库相结合,并给出实际应用实例。

本书共四篇十章,七个基础实验。第一篇为数据库基础知识,主要包括数据库基础操作和数据库中易混淆的概念。第二篇为系统开发工具与环境配置,包括数据库管理系统、Web 服务器、应用开发语言、MySQL 图形化管理工具和常用开发框架。第三篇为数据库设计与开发实操,主要包括数据库设计概述、数据库设计基本步骤和数据库设计实例。第四篇为数据库实验指导,包含七个数据库基础实验:数据库定义与基础操作、权限与安全、完整性语言、存储过程、触发器、性能优化和数据库备份与恢复。书中所举实例,都是编者从多年积累的教学经验中精选出来的,具有很强的实用性和可操作性。

另外,本书附带 1 张光盘,收录了本书涉及的源文件,便于学生和数据库研发人员高效、直观地学习。

### 图书在版编目(CIP)数据

数据库课程设计与开发实操 / 刘福江等著. —北京: 科学出版社, 2017. 5  
ISBN 978-7-03-052798-1

I. ①数… II. ①刘… III. ①数据库系统—课程设计—教材 IV. ①TP311.13-41  
中国版本图书馆 CIP 数据核字 (2017) 第 107265 号

责任编辑: 焦 健 韩 鹏 王晓丽 / 责任校对: 张小霞

责任印制: 肖 兴 / 封面设计: 铭轩堂

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>

三河市骏杰印刷有限公司 印刷

科学出版社发行 各地新华书店经销

\*

2017 年 5 月第 一 版 开本: 787×1092 1/16

2017 年 5 月第一次印刷 印张: 14 1/4

字数: 338 000

定价: 78.00 元 (含光盘)

(如有印装质量问题, 我社负责调换)

## 作者名单

刘福江 张传维 孙斌 扈震

杨林 林伟华 郭艳 万林

储菁 李珊珊 刘振等

# 前言

## 数据库

数据库技术作为数据管理的最有效手段，极大地促进了计算机应用的发展，已经成为信息基础设施的核心技术和重要基础。

目前市场上关于数据库的书籍数不胜数，有的书籍过分强调设计，甚至对需求分析还没介绍清楚，就开始直接进入设计阶段。有的书籍过分强调实际操作，设计部分一笔带过，只强调代码的编写。对于初学者，很难找到符合自身学习情况和节奏的教材；并且环境搭建、配置也是学习数据库的重要环节。因此，为了避免读者盲目地去学习数据库设计，本书权衡了设计与实操的轻重，在两者之间达成平衡。另外，在光盘里详细给出了本书的数据库设计所有需要使用的软件及插件。本书的最终目标是帮助读者，使其能够独立完成一个项目，形成一个完整系统的思路，并掌握 Web 端到接口，再到数据库连接的一套完整的操作。

## 方法

通过上机实验，帮助读者迅速掌握 SQL 语法，巩固理论知识。因此，要学好这门课程，需要通过上机实验来辅助学习，促进对理论知识的掌握，最终真正达到学为所用的目的。希望读者充分利用实验条件，认真完成实验，从实验中得到应有的锻炼和培养。

## 本书

本书选择的实验环境是 SQL Server 2008 R2、MySQL、PHP、Java、SQLite，涵盖了七个实验共 24 个课时的实验内容，具体分为：数据库定义与基础操作、权限与安全、完整性语言、存储过程、触发器、性能优化和数据库备份与恢复。实验项目包含对数据库原理的理解和运用，融合了实际的数据库编程技术，达到了理论与实践相结合、基础理解实验与综合设计实验相结合的不同层次的要求。

### 1. 从基础到实战，一步一个脚印

本书设计为：基础知识+基础实验+案例设计，既符合循序渐进的学习规律，又力求贴近项目实战等实际应用。基础知识是必备内容，读者可以在基础实验中进一步巩固基础知识。在案例设计中，使用 Java、PHP 和 MySQL、SQLite 当前较热门的语言和数据库，让内容更加贴近实际应用，让读者在学习基础知识的同时，动手进行项目的开发，对数据库设计有一个全面的认识。

### 2. 易学易用

颠覆传统的“看”书观念，变成一本能“操作”的书。本书从数据库上机的基础性



操作，再到实际应用的设计，循序渐进，易学易用，帮助读者很好地掌握数据库设计。

### 3. 贴心周到

光盘含有本书设计的源代码和环境配置说明，另外，数据库设计所需要的软件都打包放在光盘里，以免读者走弯路。同时，为了进一步方便交流学习，读者可直接发送邮件到邮箱：databaselearning@163.com。

### 致读者

亲爱的读者，感谢您在茫茫书海中选择了本书，希望它能架起你我之间学习、友谊的桥梁。编者从事数据库教学十余年，发现目前市场上图书缺少对于深奥的理论知识的充分解释，对于实际操作的解释也不够到位。因此，编者基于目前市场上的情况，紧跟数据库发展的趋势，特地编写本书。

希望本书能够帮助您解决学习或设计数据库时遇到的一些问题。由于编者水平有限，书中难免有不足之处，敬请批评指正。

作者

2017年4月

# 目 录

## 第一篇 数据库基础知识

|                              |    |
|------------------------------|----|
| 第1章 数据库基础操作 .....            | 3  |
| 1.1 添加操作 .....               | 3  |
| 1.1.1 新建数据库 .....            | 3  |
| 1.1.2 数据库中新建表 .....          | 3  |
| 1.1.3 数据库表中添加数据 .....        | 4  |
| 1.2 修改操作 .....               | 4  |
| 1.3 查询操作 .....               | 4  |
| 1.3.1 数据库/表信息查询 .....        | 4  |
| 1.3.2 单表查询 .....             | 5  |
| 1.3.3 分组查询 .....             | 6  |
| 1.3.4 连接查询 .....             | 6  |
| 1.3.5 视图查询 .....             | 7  |
| 1.4 删除操作 .....               | 7  |
| 1.5 其他 .....                 | 8  |
| 1.5.1 数据完整性 .....            | 8  |
| 1.5.2 触发器 .....              | 8  |
| 1.5.3 存储过程 .....             | 8  |
| 1.5.4 数据库恢复 .....            | 9  |
| 1.5.5 并发控制 .....             | 9  |
| 第2章 数据库中易混淆的概念 .....         | 10 |
| 2.1 DB、DBS、DBMS 和数据库技术 ..... | 10 |
| 2.2 关系、关系模式和关系数据库 .....      | 11 |
| 2.3 C/S、B/S 和两层 C/S .....    | 11 |
| 2.4 数据完整性和安全性 .....          | 12 |
| 2.5 记录、字段、属性和元组 .....        | 12 |
| 2.6 数据字典 .....               | 12 |
| 2.7 并发和并行 .....              | 13 |

## 第二篇 系统开发工具与环境配置

|                             |    |
|-----------------------------|----|
| 第3章 数据库管理系统 .....           | 17 |
| 3.1 MySQL 安装 .....          | 17 |
| 3.1.1 MySQL 软件获取 .....      | 17 |
| 3.1.2 MySQL 软件安装 .....      | 17 |
| 3.2 SQL Server 安装 .....     | 23 |
| 3.2.1 SQL Server 软件获取 ..... | 23 |
| 3.2.2 SQL Server 软件安装 ..... | 23 |
| 第4章 Web 服务器 .....           | 35 |
| 4.1 Apache 服务器 .....        | 35 |
| 4.2 IIS 服务器 .....           | 38 |
| 4.3 Tomcat 服务器 .....        | 40 |
| 第5章 应用开发语言 .....            | 42 |
| 5.1 Java .....              | 42 |
| 5.1.1 介绍 .....              | 42 |
| 5.1.2 安装和环境变量设置 .....       | 42 |
| 5.2 PHP .....               | 48 |
| 5.2.1 介绍 .....              | 48 |
| 5.2.2 安装和环境变量设置 .....       | 49 |
| 第6章 MySQL 图形化管理工具 .....     | 51 |
| 6.1 Workbench .....         | 51 |
| 6.2 Navicat .....           | 52 |
| 第7章 常用开发框架 .....            | 56 |
| 7.1 MVC 框架 .....            | 56 |
| 7.1.1 M 层 (模型) .....        | 56 |
| 7.1.2 V 层 (视图) .....        | 56 |
| 7.1.3 C 层 (控制器) .....       | 57 |
| 7.2 SSH 框架 .....            | 57 |
| 7.2.1 Struts 框架 .....       | 57 |
| 7.2.2 Spring 框架 .....       | 57 |
| 7.2.3 Hibernate 框架 .....    | 58 |
| 7.3 PHP 框架 .....            | 59 |
| 7.3.1 ThinkPHP 框架 .....     | 59 |
| 7.3.2 国外著名框架 .....          | 59 |

## 第三篇 数据库设计与开发实操

|                               |     |
|-------------------------------|-----|
| 第8章 数据库设计概述 .....             | 63  |
| 8.1 数据库设计定义 .....             | 63  |
| 8.2 数据库设计目标及意义 .....          | 63  |
| 8.2.1 设计意义 .....              | 63  |
| 8.2.2 设计目标 .....              | 64  |
| 8.3 数据库设计技巧及注意事项 .....        | 64  |
| 8.3.1 设计技巧 .....              | 64  |
| 8.3.2 注意事项 .....              | 65  |
| 8.4 课程设计规范 .....              | 65  |
| 8.4.1 命名规范 .....              | 65  |
| 8.4.2 报告形式 .....              | 66  |
| 实习题目 .....                    | 67  |
| 第9章 数据库设计基本步骤 .....           | 68  |
| 9.1 需求分析 .....                | 68  |
| 9.1.1 需求说明 .....              | 68  |
| 9.1.2 数据字典 .....              | 69  |
| 9.1.3 数据流图符号说明 .....          | 70  |
| 9.2 概念结构设计 .....              | 70  |
| 9.2.1 特点 .....                | 70  |
| 9.2.2 分类方法 .....              | 70  |
| 9.2.3 E-R图 .....              | 72  |
| 9.2.4 视图集成 .....              | 72  |
| 9.2.5 验证概念结构 .....            | 73  |
| 9.3 逻辑结构设计 .....              | 73  |
| 9.4 物理结构设计 .....              | 75  |
| 9.5 数据库实施 .....               | 75  |
| 9.6 数据库运行和维护 .....            | 75  |
| 第10章 数据库设计实例 .....            | 77  |
| 10.1 病房管理系统 (MySQL+PHP) ..... | 77  |
| 10.1.1 需求分析 .....             | 77  |
| 10.1.2 概念设计 .....             | 85  |
| 10.1.3 逻辑结构设计 .....           | 91  |
| 10.1.4 物理设计 .....             | 96  |
| 10.1.5 系统实施与系统维护 .....        | 103 |
| 10.1.6 系统运行结果 .....           | 105 |

|                                          |     |
|------------------------------------------|-----|
| 10.1.7 分模块源代码 .....                      | 114 |
| 10.1.8 ThinkPHP 框架版本 .....               | 116 |
| 10.2 食堂订餐系统 (SQL Server+Java) .....      | 117 |
| 10.2.1 需求分析 .....                        | 117 |
| 10.2.2 概念设计 .....                        | 122 |
| 10.2.3 逻辑结构设计 .....                      | 124 |
| 10.2.4 物理设计 .....                        | 126 |
| 10.2.5 系统实施与系统维护 .....                   | 129 |
| 10.2.6 系统运行结果 .....                      | 129 |
| 10.2.7 分模块源代码 .....                      | 134 |
| 10.3 家庭理财系统 (Android: SQLite+Java) ..... | 135 |
| 10.3.1 需求分析 .....                        | 135 |
| 10.3.2 概念设计 .....                        | 139 |
| 10.3.3 逻辑结构设计 .....                      | 141 |
| 10.3.4 物理设计 .....                        | 142 |
| 10.3.5 系统实施与设计分析 .....                   | 143 |
| 10.3.6 系统运行结果 .....                      | 150 |
| 10.3.7 分模块源代码 .....                      | 152 |

## 第四篇 数据库实验指导

|                        |     |
|------------------------|-----|
| 实验一 数据库定义与基础操作 .....   | 157 |
| 实验 1.1 数据库定义实验 .....   | 157 |
| 实验 1.2 数据库查询实验 .....   | 163 |
| 实验 1.3 数据库更新实验 .....   | 175 |
| 实验二 权限与安全 .....        | 182 |
| 实验三 完整性语言 .....        | 185 |
| 实验四 存储过程 .....         | 190 |
| 实验五 触发器 .....          | 198 |
| 实验六 性能优化 .....         | 203 |
| 实验 6.1 数据库查询性能优化 ..... | 203 |
| 实验 6.2 数据库结构优化 .....   | 209 |
| 实验七 数据库备份与恢复 .....     | 211 |
| 实验 7.1 数据库备份实验 .....   | 211 |
| 实验 7.2 数据库恢复实验 .....   | 215 |
| 参考文献 .....             | 217 |

# 第一篇 数据库基础知识

第1章 数据库基础操作

第2章 数据库中易混淆的概念



# 第1章 数据库基础操作

目前，互联网日趋流行，前端和后台数据库交互日显重要。对于数据库设计的初学者，一个扎实的基础，对于以后的学习显得尤为重要。无论数据库发展到什么阶段，其操作的话题始终都围绕数据库的“增删改查”。本章将对数据库的核心操作“增删改查”进行简单而精炼的介绍，为后面的实际操作打下坚实的基础。

本章目标（请在掌握的内容方框前面打钩）

- ☐ 添加操作
- ☐ 修改操作
- ☐ 查询操作
- ☐ 删除操作
- ☐ 其他

## 1.1 添加操作

数据库添加操作有四种：新建数据库、数据库中新建表、数据库表中添加数据。

### 1.1.1 新建数据库

新建数据库，语法如下：

```
create database db_name;
```

例如，create database aaa;

这里新建了一个名叫 aaa 的数据库，用户之后可以对 aaa 数据库做新建表的操作。具体事例详见第四篇的实验一。

### 1.1.2 数据库中新建表

在已有数据库新建表，语法如下：

```
create table table_name(  
column_name1 colume_type1,  
column_name2 colume_type2  
);
```

例如，create table Test(  
id int(50)  
);

**注意：**新建表时至少要包含一个字段，否则会出错。

这里向 aaa 数据库中添加了 Test 表，用户之后可以对 Test 表做添加、修改字段的操作。具体事例详见第四篇的实验一。

### 1.1.3 数据库表中添加数据

数据库中的 insert 语句语法如下：

```
insert into table_name values(column_value1,column_value2,...);
```

进行特定列数据的添加还有另外一种形式：

```
insert into table_name set column_name1 = value1,column_name2 = value2,...;
```

第一种方法将列名和列值分开了，在使用时，列名必须和列值一致。具体实例详见第四篇的实验一。

## 1.2 修改操作

数据库修改操作的本质是修改数据库的结构和字段内容，当用户有修改数据库权限时，就能对数据库进行“写”操作，通过修改数据库，从而达到满足用户需求的目的。

### 1. 修改所有数据表中的数据

```
update table_name set column_name='column_value'; //字符串加单引号''
```

例如，update Test set name = '张三';

该 SQL 语句的作用是将 Test 表中名为 name 的列名全部改为“张三”。

### 2. 带条件的修改

```
update table_name set column_name = 'column_value' where condition; //非数字加单引号''
```

例如，update Test set name = '李四' where id = 1;

该 SQL 语句的作用是仅将 Test 表中 id 号为“1”的 name 改为“李四”，而其他的 name 不变。具体事例详见第四篇的实验一。

**注意：**我们在更新数据时，一定要用 where 条件，否则数据会被破坏。

## 1.3 查询操作

数据库查询操作的本质是“看”，通过查询语句查看数据库的结构、包含的表、每个表的作用等。

### 1.3.1 数据库/表信息查询

在实际操作时，有时候需要查询整个数据库（表）信息，防止因已存在相同数据库名

(表名)而导致建库(建表)失败。另外,查询整个数据库和表信息能够帮助读者对整个系统有整体性的了解,对以后的开发工作有很大帮助。具体实例详见第四篇的实验一。

### 1. 查看数据库信息

```
show databases;
```

### 2. 查看表信息

```
show table_name;
```

### 3. 查看表的结构

```
desc table_name;
```

## 1.3.2 单表查询

### 1. 查询指定字段

在系统设计中,当进行系统修改时,有时候需要找到数据库中对应的数据表的内容,然后进行相对应的修改操作。这时就需要进行单表查询操作。语法如下:

```
select column_name from table_name;
```

例如, `select id,name from Test;`

查询考试表中的编号和考试科目名称。

### 2. 使用 order by 对查询结果进行排序

语法如下:

```
select * from table_name order by column_name desc/asc;
```

例如, `select * from Test order by score desc;`

查询考试信息表中的成绩信息,并以成绩按降序排列。

### 3. 查询表中某列并修改该列的名称

语法如下:

```
select column_name as 'column_value' from table_name;
```

例如, `select name as '考试科目' from Test;`

查询 Test 表的 name 字段并将其改为“考试科目”,注意:原表中并未改变。

### 4. 单表中单一条件查询数据

语法如下:

```
select column_name from table_name where condition; //condition 是限制条件
```

例如, `select name from Test where expense>50;`

查询考试科目信息表中考试费用大于 50 的考试科目。具体事例详见第四篇的实验一。

### 1.3.3 分组查询

在进行系统设计时,有时候需要查询表中特定字段的特定信息,然后取别名,有时还要进行排序操作,这时就需要进行分组查询。

#### 1. 使用 **having** 的单列分组查询

分组后条件的筛选必须使用 **having** 子句。

例如, `select id as '考试编号', name as '考试科目', avg(score) as '平均成绩' from Test having avg(score) > 60;`

查询 Test 表中成绩大于 60 的考试编号和考试科目信息。

#### 2. 使用 **order by** 的多列分组查询

注意: **order by** 必须放在所有句子的后面。

例如, `select sno, score from Test where cno = 'c3' order by score desc;`

查询选修 c3 号课程的学生学号及其成绩信息,并以分数按降序排列。具体事例详见第四篇的实验一。

### 1.3.4 连接查询

#### 1. 外连接

外连接包括左外连接和右外连接两种。

##### 1) 左外连接 (left outer join)

左外连接的结果集包括 **left outer** 子句中指定左表的所有行,而不只是连接列所匹配的行。如果左表的某行在右表中没有匹配行,则在相关联的右表结果集中所有的选择列均为空值。语法如下:

```
select * from table1 left outer join table2 where table1.id=table2.id;
```

##### 2) 右外连接 (right outer join)

右外连接是左外连接的反向连接,返回右表的所有行。如果右表的某行在左表中没有匹配行,则左表将返回空值。语法如下:

```
select * from table1 right outer join table2 where table1.id=table2.id;
```

#### 2. 内连接

内连接是比较运算符(包括 =、>、<、<>、>=、<=、!> 和 !<) 进行表间的比较操作,查询与连接条件相匹配的数据。根据比较运算符的不同,内连接分为等值连接和非等值连接两种。

下面以等值连接为例。

等值连接就是将多个表之间的相同字段作为条件进行数据查询，因此需要创建多个表进行实验。

语法如下：

```
select * from table1 inner join table2 where table1.id = table2.id;
```

其中主要的比较运算符有： $=$ 、 $>$ 、 $<$ 、 $>=$ 、 $<=$ 、 $!=$ （或 $<>$ ）等。当连接运算符为 $=$ 时，称为等值连接，否则称为非等值连接。

### 1.3.5 视图查询

在 SQL 中，视图查询是基于 SQL 语句的结果集的可视化的表。它可以从某个查询内部、某个存储过程内部，或者从另一个视图内部来进行查询。通过向视图添加函数、join 等，可以向用户精确地提交我们希望提交的数据。

创建视图查询的语法如下：

```
create view view_name as select column_name from table_name where condition;
```

## 1.4 删除操作

删除操作的本质是对数据库中的索引、表、字段以及数据库的删除，分为物理删除和逻辑删除。逻辑删除是名义上的删除，而物理删除是真正的删除。逻辑删除就是对要删除的数据打上一个删除标记，在逻辑上是数据被删除，但数据本身依然存在；而物理删除是把数据从存储结构中彻底删除。因此，数据库中删除操作有多种，分别有 delete、drop 和 truncate。

### 1. delete、drop 和 truncate 删除操作的相同点

- (1) truncate 和不带 where 子句的 delete 以及 drop 都会删除表内的数据。
- (2) drop、truncate 都是数据定义语言（DDL）语句，执行后会自动提交。

### 2. delete、drop 和 truncate 删除操作的不同点

- (1) drop：管理表的操作，主要用来删除表、索引、字段数据库，可以释放空间。
- (2) delete：单纯地删除数据表的数据，可以有 where 条件，删除内容但不删除定义，且不释放空间。
- (3) truncate：直接清空数据表，将其全部删除，比 delete 稍快，删除内容、释放空间但不删除定义。

### 3. 具体实例

删除 user 表中 name 等于“Mike”的前 6 条记录，可以使用如下的 delete 语句。

```
delete from user where name = ' Mike' limit 6;
```

删除 user 数据表，均可以使用 drop 和 truncate。