

# 基于 SQL Server 的学生学籍管理系统数据库设计

孟祥娜<sup>1</sup>, 秦玉平<sup>2</sup>

(1. 渤海大学 文理学院, 辽宁 锦州 121013; 2. 渤海大学 信息科学与工程学院, 辽宁 锦州 121013)

**摘要:** 以一所高校学生学籍管理系统的数据库设计为例, 按照软件工程中结构化的分析与设计方法, 对管理系统进行了需求分析, 在此基础上对数据库进行了概念设计和逻辑结构设计, 用 SQL Server 2000 进行了数据库物理设计与实现, 给出了学生学籍管理系统数据库设计的全过程。

**关键词:** 学籍管理; 数据库; 软件工程; SQL Server 数据库

## Design the Database of Student Status Management System Based on SQL Server

MENG Xiang-na<sup>1</sup>, QIN Yu-ping<sup>2</sup>

(1. College of Arts and Science, Bohai University, Liaoning Jinzhou 121013, China;

2. College of Information Science and Engineering, Bohai University, Liaoning Jinzhou 121013, China)

**Abstract:** In this paper we take database of college student status management system as an example, according to the structured analysis and design method in the software engineering, I analyze needs of management system. And then conceptual design and logic design of database took place on the basis, finally physical design and Implementation of database by the SQL Server 2000. This paper gave the whole process database of student status management system.

**Key words:** student status management; database; software engineering; SQL Server Database

### 1 引言

由于高校学生类别和专业的设置的多样化, 学生学籍管理历来是非常繁琐和复杂的工作。面对庞大的学生学籍信息, 如果采用人工管理实现起来必然相当困难, 在这种情况下学籍信息采用数据库进行管理变的尤为必要。学生学籍管理系统不但可以提高学生管理工作的效率, 还可以做到信息的规范管理科学统计和快速查询, 从而减少管理方面的工作量和人为的错误。

### 2 需求分析

学籍管理系统通常包括: 招生、学籍、教学计划、成绩等部分, 这里只对学籍及计划部分的实体进行分析, 具体如下。

#### 2.1 班级实体

班级实体用于保存班级信息, 其主要属性包括: 班代码、班名、计划、专业、年级、学制、年限、人数、班主任等。

#### 2.2 学籍档案实体

学籍档案实体用于保存学生的基本信息, 其主要属性包括: 学号、姓名、班级、性别、文化程度、政治面貌、民族、身份证号、住址、籍贯、学籍状态、毕业标志、电话及邮编等内容。

#### 2.3 奖励实体

奖励实体用于记录每个学生的奖励信息, 其主要属性包括: 学号、姓名、奖励日期、奖励类型、具体奖励内容等。

#### 2.4 处分实体

处分实体用于记录每个学生处分信息, 其主要属性包括: 学号、姓名、处分日期、处分类型、处分内容、解除日期等。

#### 2.5 专业总表实体

专业总表实体用于保存学校历年来开设的专业, 其主要属性包括: 专业号、专业名称、专业所属学科及门类、停开日期及标志等。

#### 2.6 教学计划实体

教学计划实体用于记录教学计划的基本信息, 其主要属性包括: 年级、专业、学制、学年、最低毕业学分, 必修课最低学分, 限选学分、选修课最低学分, 自开学分及实践环节最低学分等。

#### 2.7 课程总表实体

课程总表实体用于保存学校历年来开设过的课程, 其主要属性包括: 课程号、课程名、课程所属门类、实践环节及教学部门等。

#### 2.8 计划课程实体

计划课程实体用于记录一个专业教学计划包含的课程信息, 其主要属性有: 计划名、序号、课程性质(必修、选修、实践等)、课程名称、学分、学时、开设学期、考试学期及开课单位等。

### 3 基于 SQL Server 的学籍管理系统数据库

#### 3.1 概念设计

概念结构设计是整个数据库设计的关键, 根据前面的需

基金项目: 教育部专项研究课题资助项目 (2007110)。

作者简介: 孟祥娜 (1984-), 女, 硕士, 研究方向: 机器学习; 秦玉平 (1965-), 教授, 博士, 研究方向: 机器学习。

收稿日期: 2013-07-11

求分析，我们可以把各个实体抽象出来，设计如下满足用户需求的 ER 图。

(1) 班级与学生实体及其关系如图 1 所示。

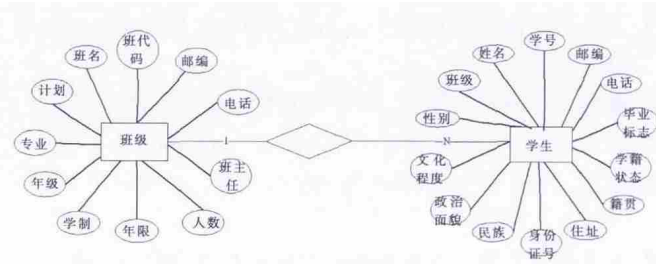


图 1 班级-学生 ER 图

一个学生可以注册一个班级，一个班级可以包含多个学生，两者之间故为一对多的关系。

(2) 学生与奖励、学生与处罚实体间的联系如图 2 所示。

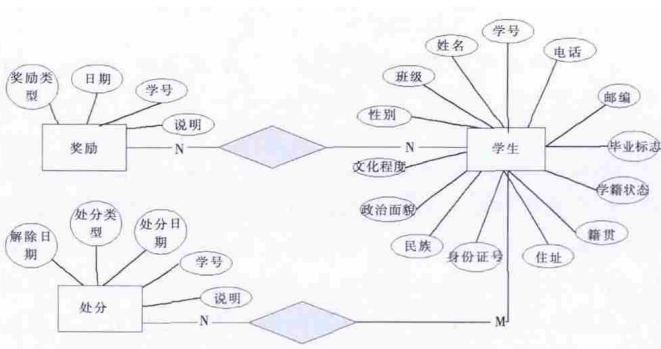


图 2 学生-奖励-处罚 E-R 图

一个学生可以得到多种奖励和处罚，一种奖励和处罚也可以多个人获得，故学生实体和奖励实体之间、学生实体和处分实体之间分别是多对多的关系。

(3) 班级与专业实体间的联系如图 3 所示。

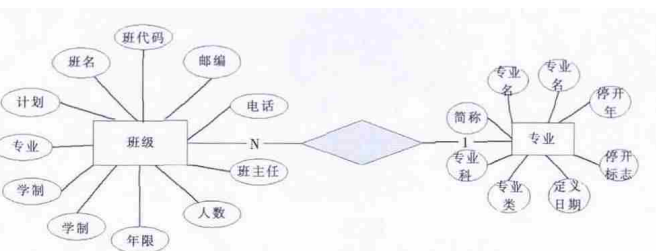


图 3 班级-专业实体间联系 E-R 图

多个班级可以属于同一专业，同班学生属于同一专业，故两者是多对一的关系。

(4) 专业与教学计划实体间的联系如图 4 所示。

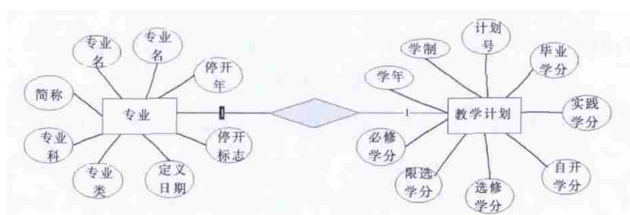


图 4 专业与教学计划 ER 图

每个专业对应惟一教学计划，一种教学计划只能让一个专业使用，故两者之间是一一对一的关系。

(5) 课程与计划课程实体间的联系如图 5 所示。

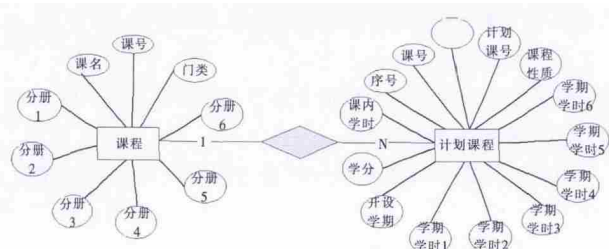


图 5 课程与教学计划 ER 图

一门课程可以是不同计划中的课程，一种计划中只能有一门该课程，故两者之间是一对多的关系。

(6) 计划课程与教学计划之间的 ER 图如图 6 所示。

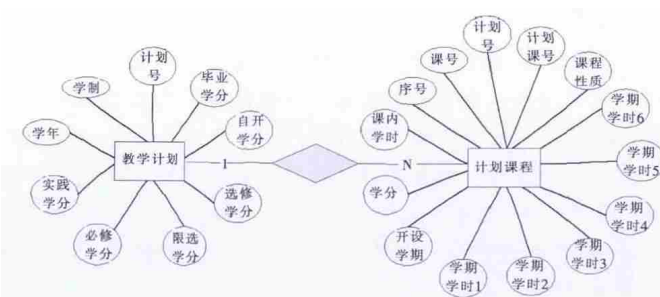


图 6 教学计划与与计划课程实体 ER 图

一个教学计划中包含多门计划课程，一门计划课程对应一个教学计划，故两者之间是一对多的关系。

(7) 以上是本系统中的基本的实体和关系，本系统是将其分步找出来加以描述，下面是学籍管理系统的总体的 ER 图，如图 7 所示。

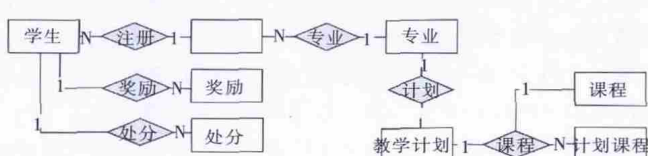


图 7 学籍管理系统 ER 图

### 3.2 数据库的物理结构设计与实现

(1) 班级信息表 (bjxx)

用于存放班级信息。包括班号、名称等 11 个字段，如图 8 所示。

列名	数据类型	长度	允许空
班号	char	14	
班名	varchar	30	
班址	varchar	12	
班址	varchar	12	
班址	char	4	
班址	char	1	
班址	varchar	5	
班址	char	8	
班址	char	12	
班址	char	6	

图 8 班级信息表

(2) 学籍档案表 (xjda)

用于存放学籍信息。包括学号、姓名等 14 个字段。

(3) 奖励信息表 (jlxx)

用于存放奖励信息。包括学号、奖励内容等 4 个字段。

(4) 处分信息表 (cfxx)

用于存放处分信息。包括学号、处分内容等 4 个字段。

(5) 专业总表 (zyzb)

用于存放专业信息。包括专业号、名称等 8 个字段。

(6) 教学计划表 (jxjhb)

用于存放教学计划信息。包括计划号、毕业学分等 9 个字段。

(7) 课程总表 (kczb)

存放课程信息。包括课程号、课程名等 9 个字段。

(8) 计划课程表 (jhkcb)

存放教学计划对应的课程。包括计划号、课号、开设学期等 14 个字段。

## 4 结语

学生学籍管理系统是典型的高校信息管理系统, 它集各种功能于一身, 方便管理学生的学籍信息, 为高校节省了管理成本, 提高了学校的工作效率。而作为学生学籍管理系统核心部分数据库, 对其进行了详细设计并实现, 但还有许多地方需要完善。

(上接第 49 页)

制作动画时就可以同步预览, 完成时可以导出视频和 GIF。

以制作一个人走路的动画为例, 通常的做法是先制作一个原地走路的影片剪辑, 然后用该影片剪辑做补间动画。但是这样做缺点很明显, 影片剪辑在舞台上无法与时间轴同步播放, 很难控制补间的时间, 因为看不到原地走路的动画节奏, 需要反复调试。而且导出 GIF 或者 AVI 的时候影片剪辑就会失效。通常只能全部在时间轴上制作, 缺点是复用这段走路动画就必须复制时间轴上的帧, 而且如果对走路动画不满意也无法只改一个元件就修改了全片所有的走路效果。这段动画如果用图形元件去做, 以上问题都可以避免。

如图 1 所示就是一个 MV 中的图形元件, 这个图形元件由男孩头发元件、女孩头发元件、男孩腿部元件、轮子元件等组成。这些元件全部使用了图形元件。所以在制作时在任何地方都能看到它每个地方运动的节奏, 导出 GIF 动画也不会出现某个地方不动的现象。

### 3.3 内在的必然联系

(1) 这 3 种元件在舞台上的实例都可以在属性面板中相互改变其行为, 也可以相互交换实例。所以说在最高层上这三者是相通的。



图 1

(2) 影片剪辑中可以嵌套另一个影片剪辑, 图形元件中也可以嵌套另一个图形元件, 但是按钮元件中不能嵌套另一个按钮元件, 3 种元件可以相互嵌套。正是因为元件之间的嵌

## 参考文献

- [1] 戴坚锋. 软件项目开发与实施 [M]. 北京: 电子工业出版社, 2009.
- [2] 吴洁明. 软件工程 [M]. 北京: 中央广播电视大学出版社, 2008.
- [3] 李继民, 李珍, 刘明, 等. 计算机专业毕业设计 (论文) 指导 [M]. 北京: 清华大学出版社, 2009 年.
- [4] 周绪. SQL Server 数据库基础教程 [M]. 北京: 清华大学出版社, 2004.
- [5] 何文华, 李萍. SQL Server 2000 应用开发教程 [M]. 北京: 电子工业出版社, 2004.
- [6] 卢越. 基于 VB.NET 的高职院校学籍管理系统的设计与实现 [J]. 电脑知识与技术, 2010, (20): 5507-5509.
- [7] 彭斌. 学籍管理系统的设计与实现 [J]. 福建电脑, 2011, (11): 125-126.
- [8] 徐萍. 高校教务管理系统的设计与实现 [D]. 南京: 南京理工大学, 2010.

套关系, 从而产生了动画中的层级关系。通过层级关系的嵌套, 能够使用相对简单的方法制作比较复杂的动画。比如制作一个人眨眼边摆头口形还在变化的动画。从元件的角度来看就是将头、眼睛、嘴巴分别制作元件, 然后分别单独给它们动作, 最后将它们组合。这几个元件的关系是头部是父级而眼睛和嘴巴是头部的子级, 随着头部的摆动, 眼睛和嘴巴也一起动。这样看来很复杂的动画, 但实现起来却很简单, 而且场景井然有序, 利用元件能够快速的管理和编辑, 适于团队操作需要, 如图 2 所示。



图 2

## 4 元件使用的重要意义

正是由于 Flash 中元件的层级嵌套和反复使用的特点, 使得用 Flash 这个软件制作动画变得简单, 也就是说 Flash 这个创作动画的工具简单好用, 从而在制作动画时不需要花很多精力在这个工具上, 而是把精力放在动画片的核心创意上即故事的创意与讲述故事的镜头语言上来, 这样可以创作出更优秀的作品。

## 参考文献

- [1] 周国栋. Flash 角色/背景/动画短片设计完全手册. 北京: 人民邮电出版社, 2009.
- [2] 智丰电脑工作室. 中文版 Flash 绘画宝典. 北京: 科学出版社, 2007.
- [3] 王俊波. Flash 动画设计与实训. 北京: 化学工业出版社, 2011.