

# 同济大学

## 《数据库系统原理》课程设计报告

### 学生宿舍信息管理系统



姓名学号 \_\_\_\_\_

专    业 \_\_\_\_\_ 计算机科学与技术 \_\_\_\_\_

指导老师 \_\_\_\_\_ 关侗红 \_\_\_\_\_

## 摘 要

随着高校学生规模的不断扩张，学生宿舍的管理工作变得越来越繁琐复杂，传统的人工管理模式已经难以满足现在的高校管理需求，为了减轻高校宿舍管理人员的负担，提高高校宿舍管理效率，利用信息化方式取代传统的人工管理方式，利用计算机完成许多传统的重复性工作，对提高管理效率与准确性有着重要的意义。所以有必要开发设计一款符合高校学生宿舍管理的系统，更好地实现学生宿舍管理事宜的统一与汇总。

学生宿舍系统的功能大大的减轻了宿舍考勤工作人员的工作量，提高了宿舍管理部门的思维运转的能力以及问题决策的分析能力。学生寝室管理部门也常常出现数字信息审核流程过长，统计时间周期延长等问题，甚至过度的工作流程让管理人员的反应能力下降。学生宿舍系统的使用，提高了工作效率，而且加大了学校的现代化水平建设，为师生创造出了更便捷的生活学习条件，也是学校整体实力提升的表现之一。

本报告以同济大学宿舍为对象，建立了学生宿舍信息管理系统。从需求与可行性分析、数据库概念设计、逻辑设计和物理设计四个方面进行了详细介绍，为后续的课程设计打下了坚实的基础。

**关键词：**学生宿舍信息系统、需求与可行性分析、概念设计、逻辑设计、物理设计

## ABSTRACT

With the continuous expansion of the size of college students, the management of student residences has become more and more complicated and complicated. The traditional manual management mode has been difficult to meet the current university management needs. In order to reduce the burden of university dormitory management personnel, improve the efficiency of college dormitory management. Using information technology to replace traditional manual management methods, using computers to complete many traditional repetitive tasks, is of great significance to improve management efficiency and accuracy. Therefore, it is necessary to develop and design a system that meets the management of university student dormitory, and better realize the unity and summary of student dormitory management matters.

The function of the student dormitory system greatly reduces the workload of the dormitory attendance staff, and improves the ability of the dormitory management department to operate and analyze the problem decision. The student bedroom management department often has problems such as excessive digital information review process, extended statistical time period, and even excessive workflows that reduce the manager's ability to respond. The use of the student dormitory system has improved work efficiency, and has increased the level of modernization of the school, creating more convenient living and learning conditions for teachers and students, and is also one of the performances of the school's overall strength.

This report establishes a student dormitory information management system for the dormitory of Tongji University. The requirements and feasibility analysis, database conceptual design, logic design and physical design are introduced in detail, which lays a solid foundation for the subsequent curriculum design.

**Key Words:** Student Dormitory Information System, Demand and Feasibility Analysis, Conceptual Design, Logic Design, Physical Design

# 目 录

一、 绪论 .....	7
1.1 课堂回顾 .....	7
1.2 本文意义 .....	7
1.3 组织结构 .....	8
二、 需求与可行性分析 .....	9
2.1 引言.....	9
2.1.1 定义.....	9
2.1.2 背景.....	9
2.1.3 编写目的.....	9
2.1.4 核心流程.....	10
2.1.5 用户需求.....	10
2.2 需求分析.....	11
2.2.1 数据字典.....	11
2.2.2 数据流图.....	14
2.3 可行性分析.....	17
2.3.1 技术可行性.....	17
2.3.2 应用可行性.....	17
2.3.3 经济可行性.....	17
三、 概念设计 .....	18
3.1 引言.....	18
3.1.1 E-R 法概述.....	18
3.1.2 概念模型的重要性.....	18
3.2 实体.....	19
3.2.1 定义.....	19
3.2.2 学生宿舍信息管理系统实体集设计.....	19
3.3 实体之间的联系.....	20
3.3.1 定义.....	20
3.3.2 学生宿舍信息管理系统联系集设计.....	20
3.4 概念结构设计方法.....	21
3.4.1 基于 E-R 法的概念模型设计.....	21
3.4.2 数据抽象.....	21
3.4.3 局部 E-R 图设计.....	21
3.4.4 综合局部 E-R 图形成总 E-R 图.....	21
3.5 实体属性局部 E-R 图.....	22
3.5.1 student 实体局部 E-R 图.....	22
3.5.2 dorm 实体局部 E-R 图.....	22

3.5.3	fee 实体局部 E-R 图	22
3.5.4	visitor 实体局部 E-R 图	23
3.5.5	manager 实体局部 E-R 图	23
3.5.6	possession 实体局部 E-R 图	23
3.6	实体联系局部 E-R 图	23
3.6.1	checkin 联系局部 E-R 图	24
3.6.2	manage 联系局部 E-R 图	24
3.6.3	visit 联系局部 E-R 图	24
3.6.4	repair 联系局部 E-R 图	25
3.6.5	quit 联系局部 E-R 图	25
3.6.6	leave 联系局部 E-R 图	25
3.6.7	check 联系局部 E-R 图	26
3.7	全局 E-R 图	26
<b>四、</b>	<b>逻辑设计</b>	<b>27</b>
4.1	引言	27
4.1.1	逻辑设计的任务和目标	27
4.1.2	规范化	27
4.1.3	范式	27
4.1.4	第一范式	27
4.1.5	第二范式	27
4.1.6	第三范式	28
4.2	关系模型设计	28
4.2.1	E-R 图向关系模型的转换规则	28
4.2.2	E-R 图向关系模型的转换	29
4.3	表结构设计	30
4.3.1	表结构设计原则	30
4.3.2	表结构设计实现	30
<b>五、</b>	<b>物理设计</b>	<b>33</b>
5.1	引言	33
5.1.1	物理设计的任务和目标	33
5.1.2	物理设计模式	33
5.1.3	使用上述四种模式的一般规则	35
5.2	事务数据访问特性	35
5.2.1	查询事务	37
5.2.2	更新事务	40
5.3	数据库索引设计	41
5.3.1	索引表	41
5.4	完整性设计	42
5.4.1	对联系的定义	42
5.4.2	对约束的定义	42

5.5 安全性设计 .....	45
5.5.1 用户管理 .....	47
5.5.2 用户权限 .....	47
<b>六、 结论与展望 .....</b>	<b>48</b>
6.1 结论 .....	48
6.2 展望 .....	48
6.2.1 信息化校园管理启动了学生的便利生活 .....	48
6.2.2 优化数据处理功能 .....	48
6.2.3 系统性能兼容问题 .....	49
<b>七、 系统实现 .....</b>	<b>50</b>
7.1 概述 .....	50
7.2 首页 .....	50
7.3 系统管理员 .....	51
7.4 宿舍管理员 .....	54
7.5 学生 .....	55
7.6 系统测试 .....	56
7.6.1 测试的目的与目标 .....	56
7.6.2 测试方法 .....	56
7.6.3 测试用例 .....	57
7.6.4 测试结论 .....	58
<b>八、 心得感想 .....</b>	<b>59</b>
<b>九、 参考文献 .....</b>	<b>60</b>

## 一、绪论

### 1.1 课堂回顾

通过一学期的数据库课程学习，在老师的谆谆教导下，我们学习了 Relational Model and Relational Algebra、SQL、ER Model、Database Design 等知识。为了能够将所学知识融会贯通，我们可选择各自感兴趣的主题进行课程设计，编写相关数据库应用，建立起一个完备的数据库系统。此报告是对具体课程设计实现的前期准备，我选择以“学生宿舍信息管理系统”设计为研究对象，讨论在建设过程中普遍存在的问题，研究分析系统涉及到各项应用和意义，通过对需求与可行性分析、概念设计、逻辑设计和物理设计进行详细阐述，为后续的系统开发实现奠定了良好的基础。

### 1.2 本文意义

如今的高校寝室的信息化管理主要是以先进信息设备，移动通讯网络以及路由器为工具，运用这些高技术产成品与学校信息系统内部的信息资源以及载体设备相结合实现有效的融合，这样的设计不仅仅是对高校宿舍信息化管理的热爱，也是提高校园服务型管理提供了客观依据。这样一个系统的建设带了以下几个方面影响：

首先，进行这样的研究是针对校园数字化信息办公高效率的基本路线。比如高校为了应对扩招均衡学生数量在各地区开设分校区，各个校区的宿舍管理部门都有着各自的信息管理系统，信息的汇报，待审核工作可以在自己的系统内部完成。宿舍管理系统的设计动力来源于计算机等信息技术为学生带来的多项便利，也是吸取了系统可以解决数据的重叠以及冗长的优势。

其次，系统的功能大大的减轻了宿舍考勤工作人员的工作量，提高了宿舍管理部门的思维运转的能力以及问题决策的分析能力。学生寝室管理部门也常常出现数字信息审核流程过长，统计时间周期延长等问题，甚至过度的工作流程让管理人员的反应能力下降。我们通过对收录信息把握寝室管理的各个环节，系统会自动进行数据的统计并且对其进行有效的整合归类，择优汰劣，最后生成一份总结报告，宿舍管理系统不仅结合了数据的统一，不同的处理方式大幅度的减少了系统数据管理的统计时间。

最后是系统对数字信息共同建立和分享功能的普及。在学校信息管理体系中，相关的高校信息包括公共发布的信息以及其他变更信息都是通过统一的网站平台系统展现给师生以及调配员。比如相关办公设备的损坏导致的数量变更信息，借用，维修情况，宿舍水电的支出管理以及数据信息汇总报告等。该研究很大程度上调整了校园信息共享的规范性，避免了信息堵塞造成的问题。

## 1.3 组织结构

第一章绪论。介绍了学生宿舍信息管理系统的研究背景意义。

第二章需求与可行性分析。给出了学生宿舍信息管理系统的定义，简述了流程，给出了数据字典、数据流图并从技术、应用和经济的角度分析了可行性。

第三章概念设计。对 E-R 法和概念模型进行了介绍，采用 E-R 模型对学生宿舍信息系统中的实体集和联系集进行了定义和设计，绘制了实体属性局部 E-R 图以及实体联系局部 E-R 图，并最终汇总得到全局 E-R 图。

第四章逻辑设计。将第三章设计好的 E-R 图向关系模型转换，详细介绍了相应的关系模式和约束，并设计实现了表结构。

第五章物理设计。在明确物理设计任务和目标的情况下，分析给出了数据库的事务数据访问特性，对一些特定的表进行了数据库索引设计，对整个数据库系统进行了安全性设计。

第六章结论与展望。



## 二、需求与可行性分析

### 2.1 引言

#### 2.1.1 定义

一个学生宿舍信息管理系统,为信息存储量比较大的学校提供了一个方便快捷的操作方式,具有运行速度快、安全性高、稳定性好等优点。

#### 2.1.2 背景

学生宿舍管理系统对于一个学校来说是必不可少的组成部分。手工记录对于规模小的学校来说还勉强可以接受,但在现代的高校宿舍管理工作中,由于高校扩招,学生人数剧增,传统的管理方法已经不能适合高校宿舍管理的需要。对于学生信息量比较庞大,需要记录存档的数据比较多的高校来说,人工记录是相当麻烦的。而且当查找某条记录时,由于数据量庞大,还只能靠人工去一条一条的查找,这样不但麻烦还浪费了许多时间,效率也比较低。当今社会是飞速进步的世界,原始的记录方式已经被社会所淘汰了,计算机化管理正是适应时代的产物。信息世界永远不会是一个平静的世界,当一种技术不能满足需求时,就会有新的技术诞生并取代旧技术。以前使用的管理系统相对落后,随着学生人数的不断增加,该系统在数据管理方面有一些混乱,越来越不能适应管理的需要,因此需要一个功能更完善,操作更方便,容量更大的管理信息系统。21 世纪的今天,信息社会占着主流地位,计算机在各行各业中的运用已经得到普及,自动化、信息化的管理越来越广泛应用于各个领域。

宿舍信息管理系统是典型的信息管理系统(MIS),其开发主要包括后台数据库的建立和维护以及前端应用程序的开发两个方面。对于前者要求建立起数据一致性和完整性强、数据安全性好的库。而对于后者则要求应用程序功能完备,易使用等特点。

本宿舍信息管理系统的一些基本功能和组成情况包括系统的需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施、数据库运行维护等步骤。

#### 2.1.3 编写目的

针对背景介绍中提到的问题,设计了一套学生宿舍管理系统。

学生宿舍管理系统采用的是计算机化管理,系统做的尽量人性化,使用者会感到操作非常方便。本宿舍管理信息系统强化了学生管理的职能,涵盖了学生管理、床位管理等主要功能,特别是对学生宿舍管理工作进行了提炼和概括,使学生宿舍管理工作日益规范化、制度化和科学化,从而达到宿舍管理高效率的目的。

本系统主要包括三大功能：录入，查询和修改。录入时最基本的系统功能，实现了所有学生情况的录入，保证了学校学生住宿情况的完整性。用户可以通过适当的查询条件，对所有或者个别要了解的学生情况进行分类的查询，系统的查询能够一步到位，还支持模糊查询，从而减小了在不知道学生具体情况下查询的难度。修改则是集查询、删除和修改功能为一体的一个管理模块，本系统可通过多个查询条件得出所要的学生纪录，并在此基础上可对学生的基本住宿资料进行修改。在简化了管理人员操作的同时，又能保证宿舍各项信息的完备以及对信息操作的快速性。

2.1.4 核心流程

- 1. 用户系统登录信息：包括添加用户，修改密码，退出系统等方面，这样可以方便管理者对宿舍楼的管理，提高查询效率；
- 2. 学生基本信息：包括 学号、学生姓名、学生性别、学生所学专业、宿舍号、入住时间等方面的信息，可以方便学生信息的查询和更新；
- 3. 宿舍基本信息：宿舍基本信息包括宿舍号、宿舍电话、住宿费、宿舍财产；
- 4. 宿舍卫生检查信息：包括宿舍号、检查情况、检查日期；
- 5. 宿舍水电缴费信息：包括宿舍号、缴费人、缴费量、缴费日期等四方面的信息；
- 6. 宿舍公物报修信息：报修信息包括宿舍号、报修人、保修日期、修理日期、报修情况等信息，具体的数据项见数据字典；
- 7. 来访登记信息：包括宿舍号、来访人、被访人、来访时间、离开时间、备注等信息。

2.1.5 用户需求

表 1.1 用户需求表

需求	描述
新生登记	新生入校，想要登记入学生的姓名学号、院系、宿舍号等基本信息
物品报修	宿舍内有物品损坏，想要报修并记录下报修时间、修复时间

卫生检查	宿舍内会有例行卫生检查，宿舍管理员想要根据检查结果登记情况
水电记录	想知道自己宿舍的水电使用情况，并据此缴费
退宿记录	学生因事想要退出宿舍
离校记录	学生因假期等原因要暂时离开宿舍
来访管理	有非本宿舍楼人员想要拜访宿舍

## 2.2 需求分析

### 2.2.1 数据字典

表 2.1 住宿学生数据字典

student				
属性名	字段	类型	长度	约束
学号	Sno	integer		主键，not null
姓名	Sname	varchar2	20	not null
性别	Sgender	varchar2	4	‘female’ 或 ‘male’，not null
专业	Sdept	varchar2	40	not null
宿舍号	Dno	varchar2	10	外键，not null
入住时间	Scheckin	date	8	not null

表 2.2 学生宿舍数据字典

dorm				
属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	主键，not null
宿舍电话	Dphone	varchar2	15	not null
物品编号	Pno	integer		外键，not null

表 2.3 报修信息数据字典

**repair**

属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	外键，not null
提交日期	Rsubmit	date	8	not null
解决日期	Rsolve	date	8	not null
报修原因	Rreason	varchar2	50	not null
物品编号	Pno	integer		外键，not null

表 2.4 卫生检查信息数据字典

**check**

属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	外键，not null
检查时间	ChTime	date	8	not null
检查结果	ChResult	varchar2	10	not null

表 2.5 水电信息数据字典

**fee**

属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	外键，not null
已用费用	Fcost	integer		not null
剩余费用	Fleftfee	integer		not null
续交费用	Fctfee	integer		not null

表 2.6 来访人员数据字典

**visitor**

属性名	字段	类型	长度	约束
姓名	Vname	varchar2	20	not null
性别	Vgender	varchar2	4	‘female’ 或 ‘male’，not null

进入时间	Vin	timestamp	20	not null
离开时间	Vout	timestamp	20	not null

表 2.7 离校数据字典

**leave**

属性名	字段	类型	长度	约束
学号	Sno	integer		外键, not null
姓名	Sname	varchar2	20	外键, not null
离校时间	Lout	timestamp	20	not null
返校时间	Lin	timestamp	20	not null

表 2.8 退宿数据字典

**quit**

属性名	字段	类型	长度	约束
学号	Sno	integer		外键, not null
姓名	Sname	varchar2	20	外键, not null
退宿时间	Qout	time	8	not null
退宿原因	Qreason	varchar2	40	not null

表 2.9 宿舍管理员数据字典

**manager**

属性名	字段	类型	长度	约束
管理员 ID	Mno	integer		主键, not null
管理员密码	Mpassword	varchar2	20	not null

表 2.10 宿舍财产数据字典

**possession**

属性名	字段	类型	长度	约束
物品编号	Pno	integer		主键, not null
物品名	Pname	varchar2	20	not null

### 2.2.2 数据流图

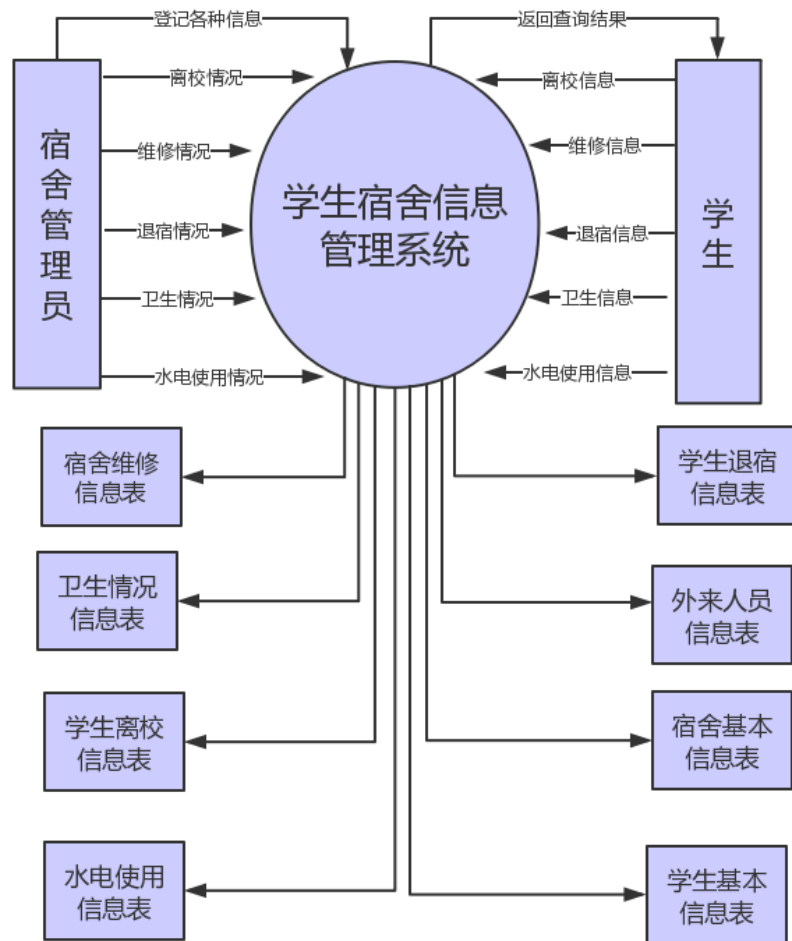


图 2.1 学生宿舍信息管理系统顶层数据流图

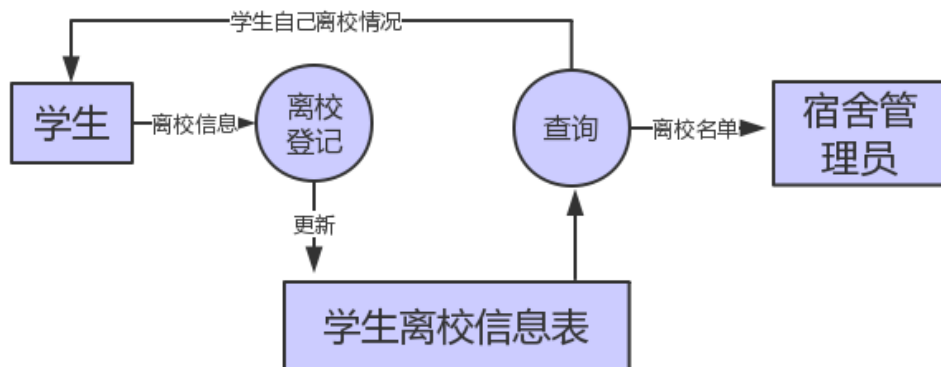


图 2.2 学生离校子数据流图

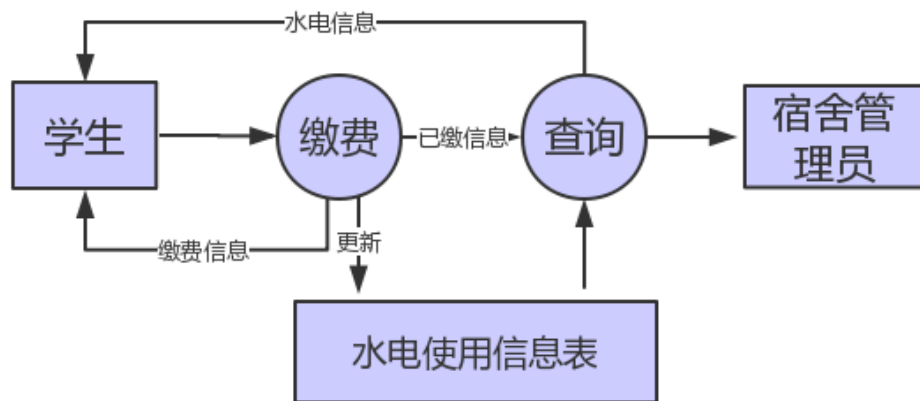


图 2.3 宿舍水电使用情况子数据流图

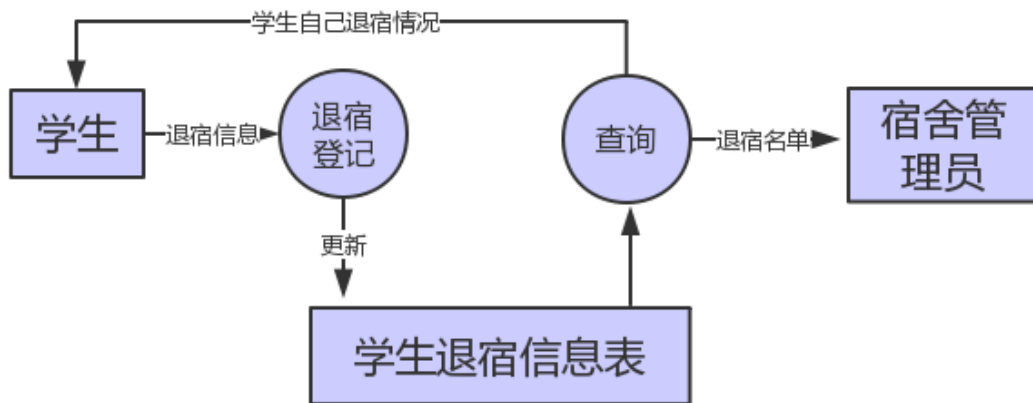


图 2.4 学生退宿子数据流图

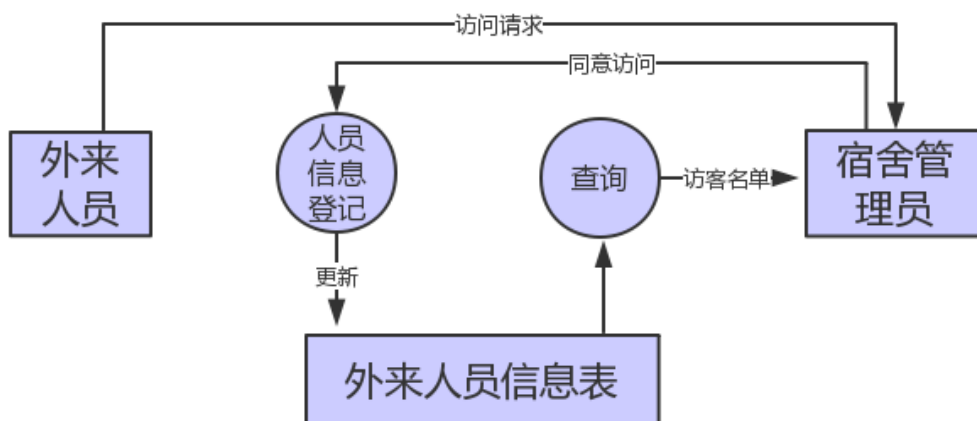


图 2.5 外来人员来访子数据流图

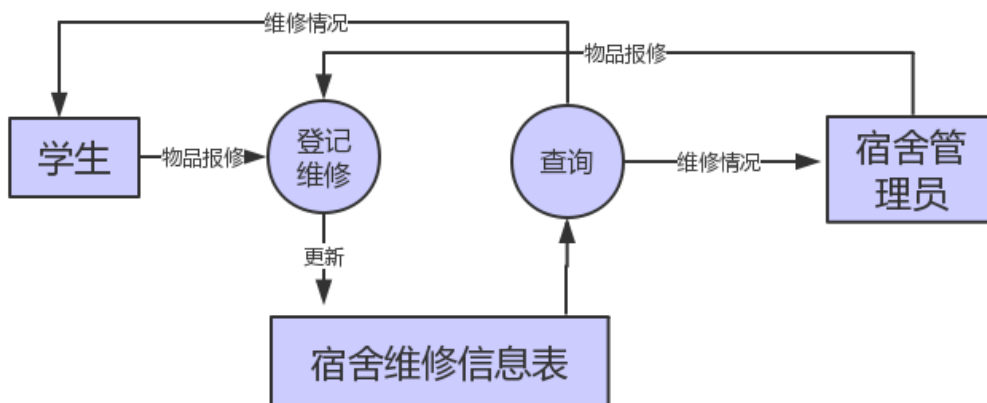


图 2.6 宿舍物品报修子数据流图

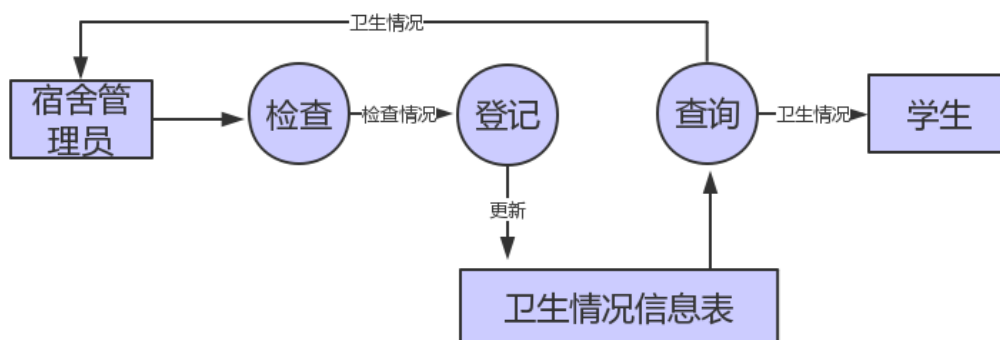


图 2.7 宿舍卫生检查子数据流图

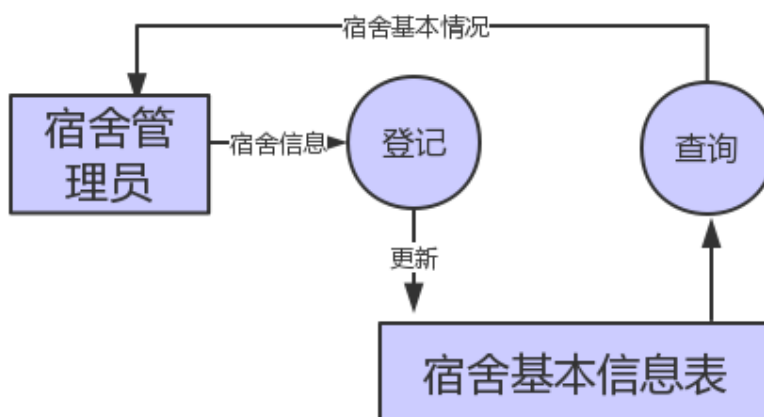


图 2.8 宿舍基本情况子数据流图



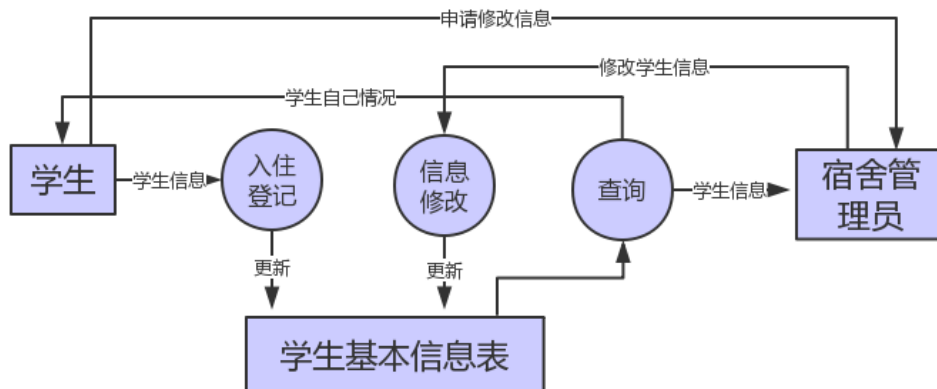


图 2.9 学生基本情况子数据流图

## 2.3 可行性分析

### 2.3.1 技术可行性

本次“宿舍信息管理系统”设计的总体任务是实现宿舍管理的系统化、规范化、自动化，从而达到宿舍管理高效率的目的。鉴于我们以前学过高级程序设计，这学期又学了数据库，所以对开发管理系统也有了一定的认识。由于本设计的最重要体现的是实用性，所以通过调查与分析，结合自己现有的技术水平加上老师的指导，能按期完成本次课程设计。

### 2.3.2 应用可行性

本系统是为学校的宿舍管理者开发的，所有本系统的用户对象是宿舍的管理人员。由于本软件设计简单，附有详细的使用说明，系统的研制和开发充分考虑了用户的工作流程且界面直观易于学习。系统兼备了以上的优点，用户只需懂得简单的计算机操作知识，就能自由应用本软件。

### 2.3.3 经济可行性

信息化管理可以减少人力成本以及减轻后勤管理团队的建设成本。学生宿舍楼管理办公室配备计算机一台，基于较为单一的管理计算机硬软件成本也比较低，并且有现成的校园网络故可以忽略信息系统的网络成本。

## 三、概念设计

### 3.1 引言

#### 3.1.1 E-R 法概述

E-R 法（Entity-Relation Approach，实体联系方法）也称为 E-R 模型；

E-R 法的思想是用 E-R 图描述现实世界的信息，这种信息结构称为概念结构，然后根据具体系统的要求将概念结构转换成特定系统所能接受的逻辑结构。

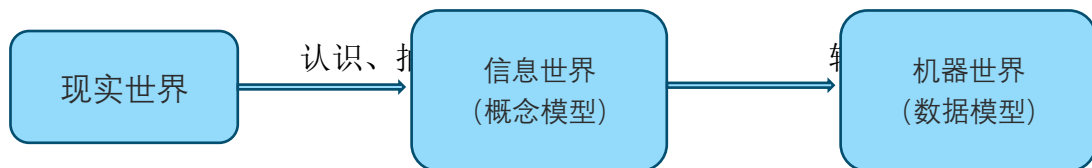
E-R 法由两部分组成：

1. 用 E-R 图描述现实世界
2. 将 E-R 图转换成相应的数据模型

E-R 模型中，其“联系”能够对不同实体关系进行描述。实践应用中，首先需要通过局部结构随机确定 2 个实体类型；其次是按照需求分析结果，判断实体类型间有无关系。如果有关系，那么必须明确是何种关系，比如 1:N, M:N, 1:1 等。与此同时，还需判断各实体类型内部有无关系，实体类型间有无关系，众多实体间有无关系等。当关系类型明确时，不仅要避免发生冗余关系，而且要进行命名、设置主键。

#### 3.1.2 概念模型的重要性

概念模型用于信息世界建模，是现实世界到信息世界的抽象，是用户和数据库设计人员进行交流的语言。



在完成概念设计工作的时候，核心是对本项目涉及的各种对象进行抽象处理，得到整个系统的概念模型。在具体的项目中，应当把真实世界客观对象抽象处理，从而准确地描述企业组织的信息。建立对应的概念结构后，接下来的工作就是将其转化为对应的物理结构。对于本项目所使用的数据库系统来说，因为逻辑结构、概念模型之间互不影响，两者之间彼此独立、彼此支持，概念模型能够支持数据库 DBMS。概念模型既能够真实充分地反映现实世界中事物与事物之间的联系，又能够清晰地表达概念设计结果。概念模型，简洁明确独立于机器，易于用户接受理解，便于应用人员和数据库人员进行相互交流。同时概念模型容易向层次模型、关系模型以及网状模型数据转变，概念模型有效实现了现实世界到机器世界的过渡转变。

## 3.2 实体

### 3.2.1 定义

实体是现实世界中可区别于所有其他对象的一个“事物”。每个实体有一组性质，其中一些性质的值可以唯一地标识一个实体。实体可以是实实在在的，如人或书；也可以是抽象的，如课程或机票预订等。

实体通过一组属性来表示。属性是实体集中每个成员所拥有的描述性性质。为某实体集指定一个属性表明数据库为该实体集中每个实体存储相似的信息；但每个实体在每个属性上都有各自的值。每个实体的每个属性都有一个值。

实体集是相同类型即具有相同性质（或属性）的一个实体集合。例如，实体集`student`可以表示大学中所有学生的集合。

### 3.2.2 学生宿舍信息管理系统实体集设计

当我们使用 E-R 模型设计数据库时，通常从确定那些应当包含的实体集开始。当决定好实体集后，我们必须挑选恰当的属性，这些属性要表示我们在数据库中所捕获的不同的值。一旦选择好实体和它们相应的属性，不同实体间的联系集就建立起来了。

一个好的实体-联系设计不包含冗余的属性。对于我们学生宿舍信息管理系统的例子，在下面列出实体集以及他们的属性，主码以下划线表明。

1. `student`: 包含属性 (Sno、Sname、Sgender、Sdept)
2. `dorm`: 包含属性 (Dno、Dphone)
3. `fee`: 包含属性 (Fcost、Fleftfee、Fctfee)
4. `visitor`: 包含属性 (Vname、Vgender、Vin、Vout)
5. `manager`: 包含属性 (Mno、Mpassword)
6. `possession`: 包含属性 (Pno、Pname)

### 3.3 实体之间的联系

#### 3.3.1 定义

联系是指多个实体间的相互关联。同一个实体集内部各实体之间也可以存在一对一、一对多、多对多的联系（如图1）。三个或多个实体型间可能具有联系（如图2）、两个实体型之间可具有多种联系（如图3）。

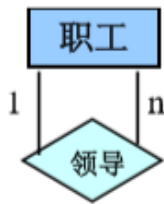


图 1

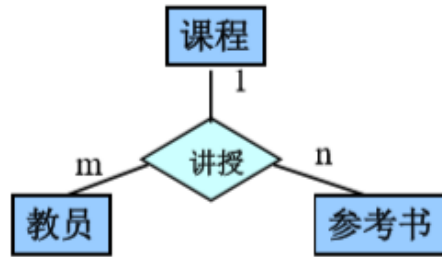


图 2

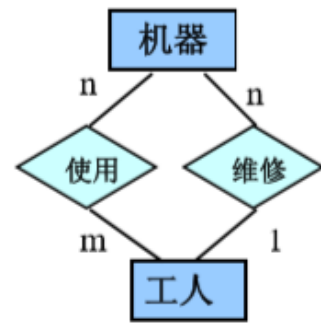


图 3

联系集是相同类型联系的集合。正规地说，联系集是 $n \geq 2$ 个（可能相同的）实体集上的数学关系。如果 $E_1, E_2, \dots, E_n$ 为实体集，那么联系集R是

$$\{(e_1, e_2, \dots, e_n) | e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

的一个子集，而 $(e_1, e_2, \dots, e_n)$ 是一个联系。

#### 3.3.2 学生宿舍信息管理系统联系集设计

我设计的联系集如下。checkin、manage、visit、repair、quit、leave、check。

1. checkin: 关联学生和宿舍
2. manage: 关联宿舍管理员和宿舍
3. visit: 关联学生和来访人员
4. repair: 关联学生和宿舍财产
5. quit: 关联学生和宿舍
6. leave: 关联学生和宿舍
7. check: 关联宿舍管理员和宿舍

### 3.4 概念结构设计方法

设计概念结构通常有四类方法：

1. 自顶向下：首先定义全局概念结构的框架，然后逐步细化
2. 自底向上：首先定义各局部应用的概念结构，然后将其集成起来
3. 逐步扩张：首先定义核心概念结构，然后向外扩充
4. 混合策略：自顶向下设计全局概念框架，自底向上设计各局部概念结构  
其中最常用的方法是自底向上方法。

#### 3.4.1 基于 E-R 法的概念模型设计

基于 E-R 法的概念模式自底向上的设计方法分为两个阶段：

1. 数据抽象与局部 E-R 图设计
2. 综合局部 E-R 图形成总 E-R 图

#### 3.4.2 数据抽象

局部 E-R 图设计的基本思想是利用数据抽象机制对需求分析阶段收集的数据进行分类、组织，形成实体模型并确定实体之间的联系类型，设计局部 E-R 图。

数据抽象机制包括：

1. 分类：定义某一概念作为现实世界中一组对象的类型。这些对象具有某些共同的特性和行为，如实体型
2. 聚集：定义某一类型的组成成分，如属性的聚集组成了实体型
3. 概括：定义类型之间的一种子集联系

#### 3.4.3 局部 E-R 图设计

局部 E-R 图具体设计方法：

1. 选择局部应用
2. 以需求分析中得到的数据元素表为基础，建立实体模型
3. 确定实体之间的联系类型。用 E-R 图表示这些实体与实体之间的联系，形成局部 E-R 图

#### 3.4.4 综合局部 E-R 图形成总 E-R 图

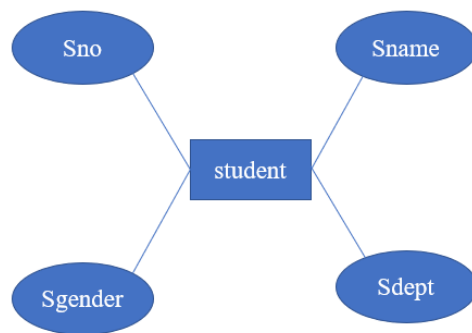
设计总 E-R 图（全局概念模式）可以有两种方法：

1. 多个局部 E-R 图一次集成
2. 逐步集成，用累加的方式一次集成两个局部 E-R 图

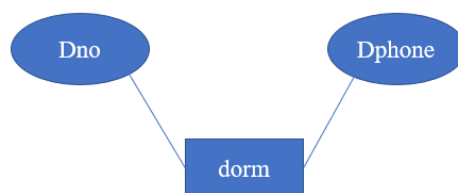
### 3.5 实体属性局部 E-R 图

根据需求分析,总结出该学生宿舍信息管理系统中的实体包括: student、dorm、fee、visitor、manager、possession。

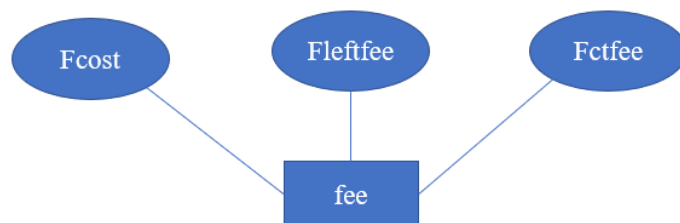
#### 3.5.1 student 实体局部 E-R 图



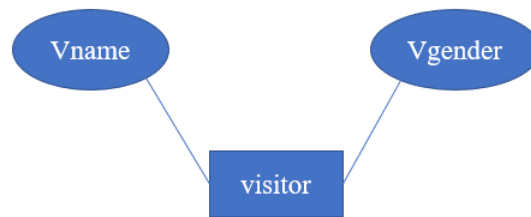
#### 3.5.2 dorm 实体局部 E-R 图



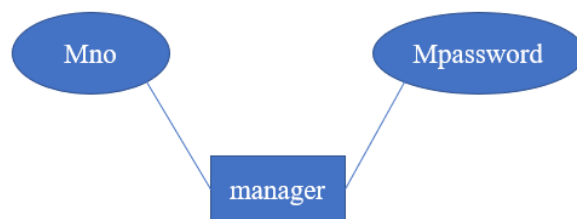
#### 3.5.3 fee 实体局部 E-R 图



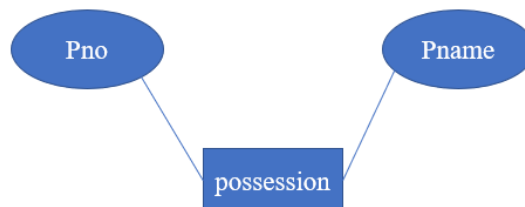
#### 3.5.4 visitor 实体局部 E-R 图



#### 3.5.5 manager 实体局部 E-R 图



#### 3.5.6 possession 实体局部 E-R 图

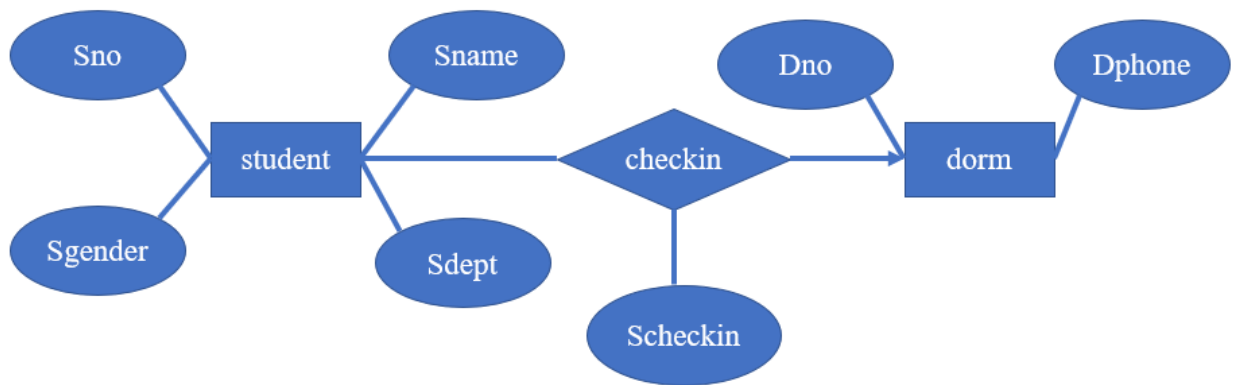


### 3.6 实体联系局部 E-R 图

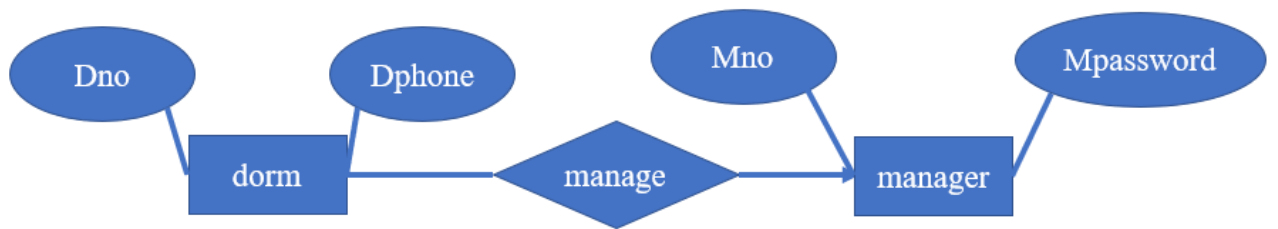
根据需求分析，总结出该学生宿舍信息管理系统中的联系包括：checkin、manage、visit、repair、quit、leave、check。

在学生宿舍信息管理系统中，数据库设计必须重点考虑如下实体：student、manager、dorm 等，通过上述实体间的关系设计不难发现，关系比较复杂，包括一对一、一对多、多对多等关系。例如一个学生一次只能住进一间宿舍，而一间宿舍可以住进多个学生，所以，这两者之间就是一种一对多的关系。在数据库设计过程中，这些一般通过 E-R 图进行描述，详情可参照下图继续了解：

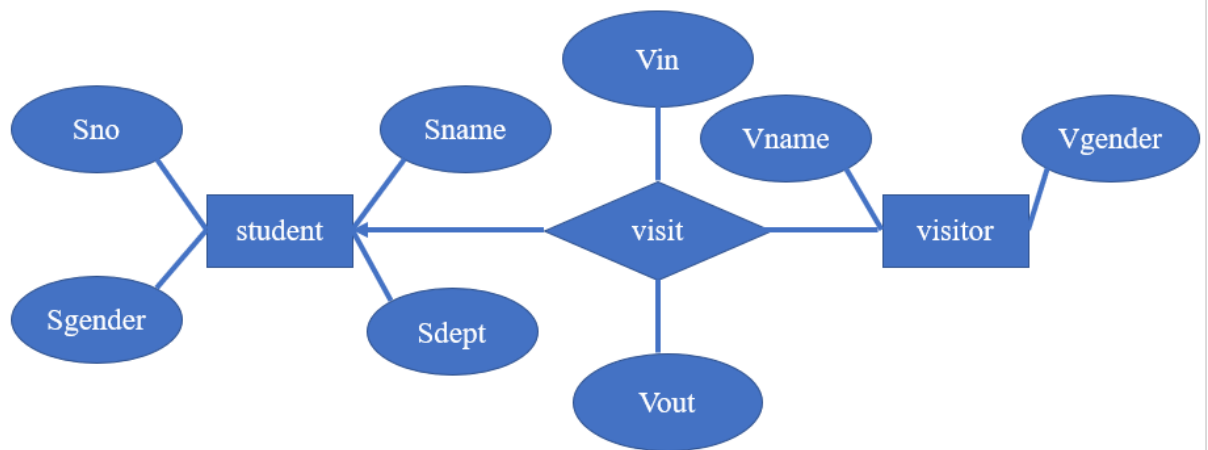
### 3. 6. 1 checkin 联系局部 E-R 图



### 3. 6. 2 manage 联系局部 E-R 图

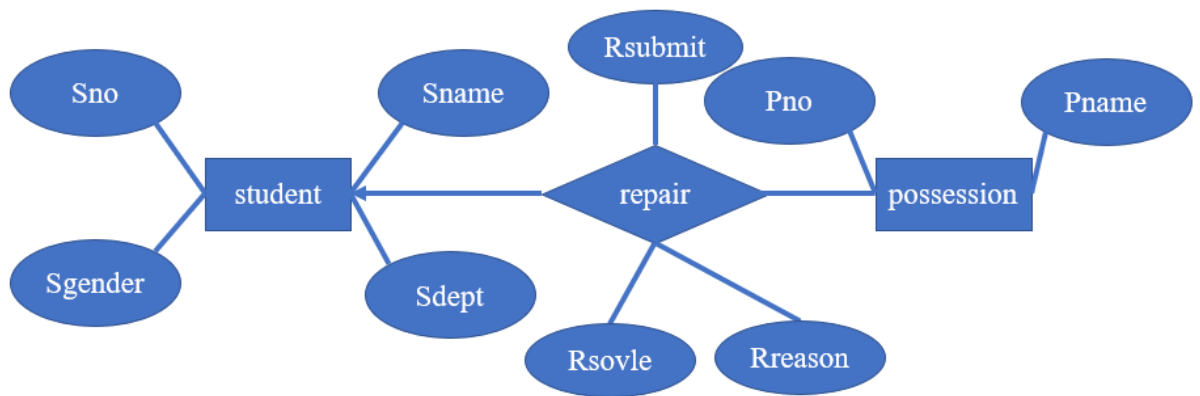


### 3. 6. 3 visit 联系局部 E-R 图

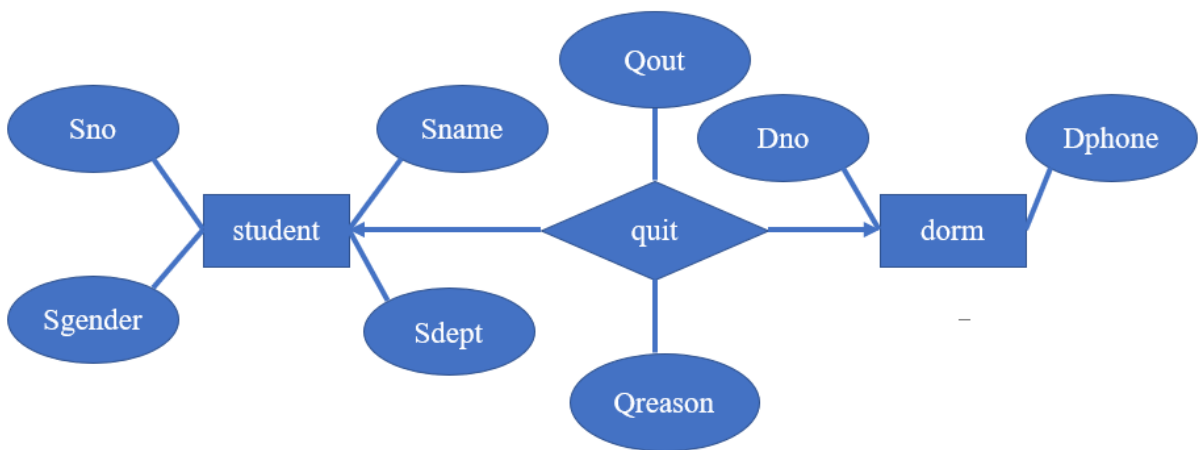




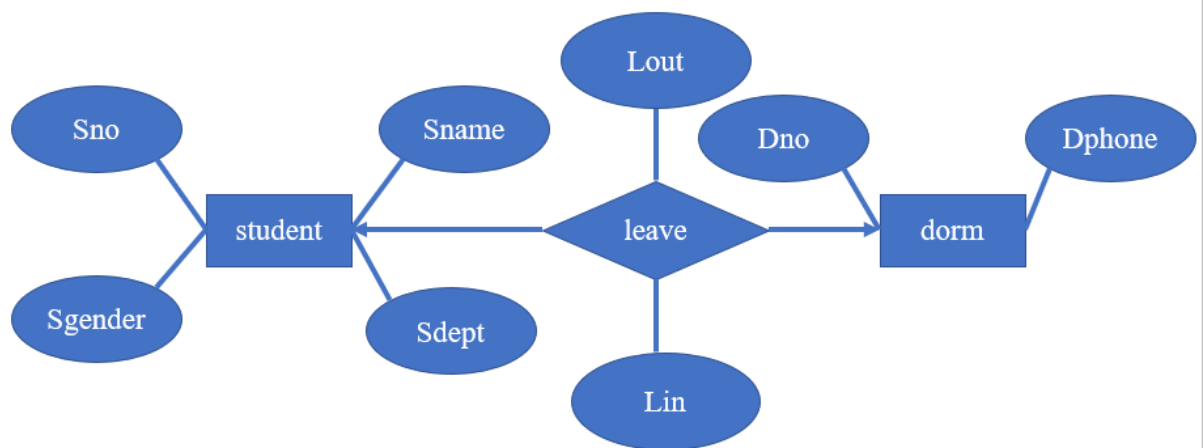
3.6.4 repair 联系局部 E-R 图



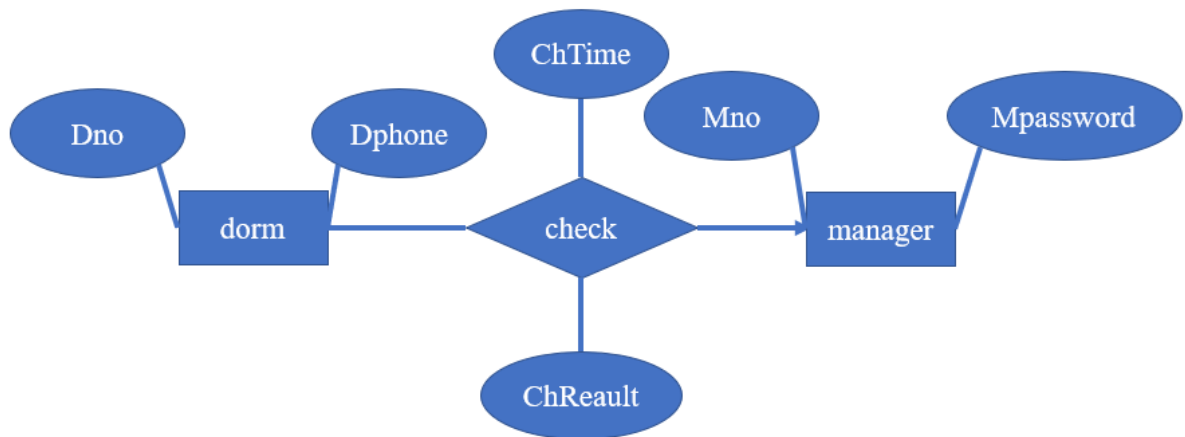
3.6.5 quit 联系局部 E-R 图



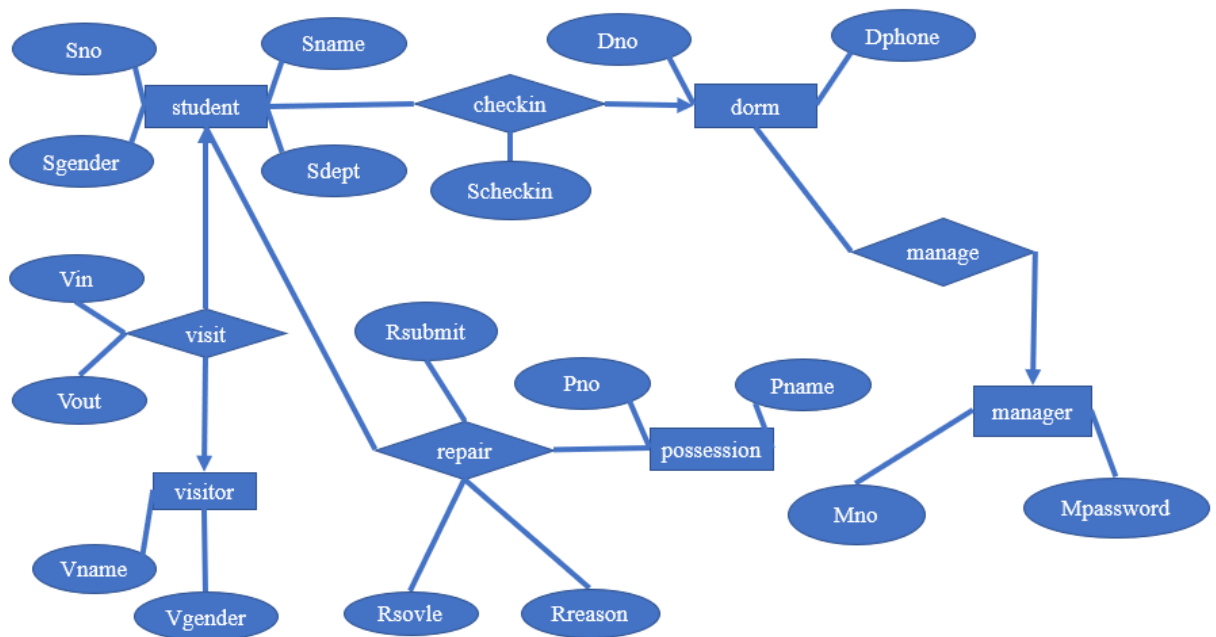
3.6.6 leave 联系局部 E-R 图



### 3.6.7 check 联系局部 E-R 图



### 3.7 全局 E-R 图



## 四、逻辑设计

### 4.1 引言

#### 4.1.1 逻辑设计的任务和目标

概念设计阶段是独立于任何一种数据模型的，但是逻辑设计阶段就与选用的 DBMS 产品发生关系了。系统逻辑设计的任务就是将概念设计阶段设计好的基本 E-R 图转化为选用 DBMS 产品所支持的数据模型相符合的逻辑结构。数据库的逻辑结构是一种抽象的概念，主要用来描述各类数据类型的关系与结构。

#### 4.1.2 规范化

一个低一级范式的关系模式，通过模式分解可以转换为若干个高级范式的关系模式的集合，这样的过程称作规范化。

#### 4.1.3 范式

如果一个关系满足某个指定的约束集，则称它属于某种特定的范式（Normal Form）；目前关系数据库有六种范式，各范式呈递次规范，越高的范式数据冗余越小。最常用的是第一到第三范式。

#### 4.1.4 第一范式

满足最低要求约束的称为第一范式，简称 1NF。当一个关系只包含原子值这一约束时，称为 1NF。原子值即为二维表的每一行和列的交叉位置上总是精确地存在的一个值，而不是值集，即不能“表中有表”。

#### 4.1.5 第二范式

如果关系模式 R 满足第一范式，并且 R 的所有非主属性都完全依赖于 R 的每一个候选关键属性，称 R 满足第二范式，简记为 2NF。

需要注意的是：

1. 如果关系 R 的全体属性都是 R 的主属性，那么 R 满足 2NF
2. 从 1NF 中消除非主属性对码的部分函数依赖，则可获得 2NF 关系
3. 在 2NF 中，允许主属性部分函数依赖于码

#### 4.1.6 第三范式

如果关系模式 R 满足第二范式，且每个非主属性都不传递依赖于 R 的任何码，称 R 满足第三范式，简记为 3NF。

### 4.2 关系模型设计

数据库设计是对数据进行组织化和结构化的过程，即关系模型的设计。

对于项目规模小、用户数量少的情况，处理数据库中的表结构相对轻松；但随着项目的发展，相应的数据库架构、关系模型表结构越来越复杂，之前写的 SQL 语句是很笨拙并且效率低下的。更可怕的是，由于表结构定义不合理，会导致对数据的增删改查不方便不高效；最致命的是，扩展性极差，不能应对业务的变化。此时对我们计算机专业学生关系模式设计能力要求提高了，我们要学习和掌握数据库的规范化流程，以指导我们更好的设计数据库的表结构，减少冗余的数据，借此可以提高数据库的存储效率，数据完整性和可扩展性。

简洁、结构明晰的表结构对数据库的设计是相当重要的。规范化的表结构设计，在以后的数据维护中，不会发生插入(insert)、删除(delete)和更新(update)时的异常。反之，数据库表结构设计不合理，不仅会给数据库的使用和维护带来各种各样的问题，而且可能存储了大量不需要的冗余信息，浪费系统资源。

关于数据库关系模型设计的问题，就是在实际项目中，我们应该构造几个关系模型，每个关系（表）由哪些属性（列）组成，不同关系之间有什么关联。

#### 4.2.1 E-R 图向关系模型的转换规则

联系转换成关系模型时，要根据联系方式的不同采用不同的转换方式。

##### 1. 1:1 联系的转换方法

- a) 将 1:1 联系转换为一个独立的关系：与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，且每个实体的码均是该关系的候选码
- b) 将 1:1 联系与某一端实体集所对应的关系合并，则需要在被合并关系中增加属性，其新增的属性为联系本身的属性与联系相关的另一个实体集的码

##### 2. 1:n 联系的转换方法

- a) 一种方法是将联系转换为一个独立的关系，其关系的属性由与该联系相连的各实体集的码以及联系本身的属性组成，而该关系的码为 n 端实体集的码

- b) 另一种方法是在 n 端实体集中增加新的属性，新属性由联系对应的 1 端实体集的码和联系自身的属性构成，新增属性后原关系的码不变

### 3. m:n 联系的转换方法

在向关系模型转换时，一个 m:n 联系转换为一个关系。转换方法为：与该联系相连的各实体集的码以及联系本身的属性均转换为关系的属性，新关系的码为两个相连实体码的组合（该码为多属性构成的组合码）。

#### 4.2.2 E-R 图向关系模型的转换

根据概念设计中的 E-R 图，可知有如下几种联系：

1. 1:1 联系：学生与退宿、学生与离校
2. 1:n 联系：宿舍与学生、宿舍管理员与宿舍、宿舍管理员与卫生检查、学生与来访人员、学生与宿舍财产

故据此可以得到具体的 E-R 图向关系模型的转换结果。

1. student 表：(Sno, Sname, Sgender, Sdept, Scheckin, Dno)
2. dorm 表 (Dno, Dphone, Pno)
3. repair 表 (Rsubmit, Rsolve, Rreason, Dno, Pno)
4. check 表 (ChTime, ChResult, Pno, Dno)
5. fee 表 (Fcost, Fleftfee, Fctfee, Dno)
6. visitor 表 (Vname, Vgender, Vin, Vout, Sno)
7. leave 表 (Lout, Lin, Sno)
8. quit 表 (Qout, Qreason, Sno)
9. manager 表 (Mno, Mpassword)
10. possession 表 (Pno, Pname, Sno)

## 4.3 表结构设计

### 4.3.1 表结构设计原则

1. 应该根据系统结构中的组件划分，针对每个组件所处理的业务进行组件单元的数据库设计，而不是针对整个系统进行数据库设计
2. 不同组件间所对应的数据库表之间的关联应该尽可能减少，为系统或表结构的重构提供可能性
3. 采用领域模型驱动的方式和自顶向下的思路进行数据库设计，首先分析系业务，根据职责定义对象
4. 一个表中的所有非关键字属性都依赖于整个关键字。关键字可以是一个属性，也可以是多个属性的集合，总之应该确保关键字能够保证唯一性
5. 应针对所以表的主键和外键建立索引，有针对性得建立组合属性的索引，提高检索效率

### 4.3.2 表结构设计实现

表 2.1 住宿学生表

student				
属性名	字段	类型	长度	约束
学号	Sno	integer		主键，not null
姓名	Sname	varchar2	20	not null
性别	Sgender	varchar2	4	‘female’ 或 ‘male’，not null
专业	Sdept	varchar2	40	not null
宿舍号	Dno	varchar2	10	外键，not null
入住时间	Scheckin	date	8	not null

表 2.2 学生宿舍表

dorm				
属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	主键，not null

宿舍电话	Dphone	varchar2	15	not null
物品编号	Pno	integer		外键，not null

表 2.3 报修信息表

**repair**

属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	外键，not null
提交日期	Rsubmit	date	8	not null
解决日期	Rsolve	date	8	not null
报修原因	Rreason	varchar2	50	not null
物品编号	Pno	integer		外键，not null

表 2.4 卫生检查信息表

**check**

属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	外键，not null
检查时间	ChTime	date	8	not null
检查结果	ChResult	varchar2	10	not null
物品编号	Pno	integer		外键，not null

表 2.5 水电信息表

**fee**

属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	外键，not null
已用费用	Fcost	integer		not null
剩余费用	Fleftfee	integer		not null
续交费用	Fctfee	integer		not null

表 2.6 来访人员表

**visitor**

属性名	字段	类型	长度	约束
-----	----	----	----	----

姓名	Vname	varchar2	20	not null
性别	Vgender	varchar2	4	‘female’ 或 ‘male’ , not null
进入时间	Vin	timestamp	20	not null
离开时间	Vout	timestamp	20	not null
学号	Sno	integer		外键, not null

表 2.7 离校表

**leave**

属性名	字段	类型	长度	约束
学号	Sno	integer		外键, not null
离校时间	Lout	timestamp	20	not null
返校时间	Lin	timestamp	20	not null

表 2.8 退宿表

**quit**

属性名	字段	类型	长度	约束
学号	Sno	integer		外键, not null
退宿时间	Qout	time	8	not null
退宿原因	Qreason	varchar2	40	not null

表 2.9 宿舍管理员表

**manager**

属性名	字段	类型	长度	约束
管理员 ID	Mno	integer		主键, not null
管理员密码	Mpassword	varchar2	20	not null

表 2.10 宿舍财产表

**possession**

属性名	字段	类型	长度	约束
物品编号	Pno	integer		主键, not null
物品名	Pname	varchar2	20	not null
学号	Sno	integer		外键, not null



## 五、物理设计

### 5.1 引言

#### 5.1.1 物理设计的任务和目标

数据库在数据库的物理设计就是为逻辑数据模型选取一个最合适用要求的权利结构的过程，在这个阶段中要完成两大任务：

1. 确定数据库的物理结构，在关系数据库中主要是存取方法和存储结构；
2. 对物理结构进行评价，评价的重点是时间和空间效率。

#### 5.1.2 物理设计模式

模式类似于定式，就是遇到反复出现的同一问题时所固定使用的解决方案。设计模式不必多，只要掌握其中关键的几个，再结合实际的业务需求，一个完整的数据库模型就可以推导出来。其中主要有四种物理设计模式，下面将逐一介绍：

##### 1. 主扩展模式

主扩展模式，通常用来将几个相似的对象共有属性抽取出来，形成一个“公共属性表”；其余属性则分别形成“专有属性表”，且“公共属性表”与“专有属性表”都是“一对一”的关系。

“专有属性表”可以看作是对“公共属性表”的扩展，两者合在一起就是对一个特定对象的完整描述，故此得名“主扩展模式”。对于主表来说，从表既可以没有记录，也可以有唯一一条记录来对主表进行扩展说明，这就是“主扩展模式”。

##### 2. 主从模式

主从模式，是数据库设计模式中最常见、也是大家日常设计工作中用的最多的一种模式，它描述了两个表之间的主从关系，是典型的“一对多”关系。

举例如下：比如论坛程序。一个论坛通常都会有若干“板块”，在每个板块里面，大家可以发布很多的新帖。这时候“板块”和“发帖”就是主从模式，主表是“板块”，从表是“发帖”，二者是“一对多”的关系。多人也可以对感兴趣的同一份发帖进行回复，以表达各自的意见，这时候，一个“发帖”就有了多份“回复”，又构成了一个“主从模式”。

##### 3. 名值模式

名值模式，通常用来描述在系统设计阶段不能完全确定属性的对象，这些对象的属性在系统运行时会有很大的变更，或者是多个对象之间的属性存在很大的差异。

举例如下：使用名值模式进行设计时，如果对“其他属性”仅作浏览保存、不作其它任何特殊处理，则通常会设计一个“属性模板”表，该表的数据记录在系统运行时动态维护。系统运行时，如需维护“产品其他属性”，可先从“属性模板”中选择一个属性名称，然后填写“属性值”保存，系统会将对应的产品 ID、属性模板 ID 及刚刚填写的“属性值”一起保存在“产品其他属性”里，这样就完成了相关设置。无论产品的其他属性需求发生怎样的变化、怎样增删改属性，都可以在运行时实现，而不必修改数据库设计和程序代码。

使用名值模式进行设计时，如果对“其他属性”有特殊处理，比如统计汇总，那么这个属性名称需要在程序代码中作“硬编码”，即该属性名称需要在程序代码中有所体现，此时可以在“产品其他属性”表中直接记录“属性名称”，不再需要“属性模板”表。

系统运行时，如需维护“产品其他属性”，程序直接列出“属性名称”，然后填写“属性值”保存，系统会将对应的产品 ID、属性名称及刚刚填写的“属性值”一起保存在“产品其他属性”里，这样就完成了相关设置。以后如果需求发生变更，则只需修改相应的程序代码即可，不必修改数据库设计。

#### 4. 多对多模式

多对多模式，也是比较常见的一种数据库设计模式，它所描述的两个对象不分主次、地位对等、互为一对多的关系。对于 A 表来说，一条记录对应着 B 表的多条记录，反过来对于 B 表来说，一条记录也对应着 A 表的多条记录，这种情况就是“多对多模式”。

“多对多模式”需要在 A 表和 B 表之间有一个关联表，这个关联表也是“多对多模式”的核心所在。根据关联表是否有独立的业务处理需求，可将其划分为两种细分情况。

关联表有独立的业务处理需求。举例如下：比如网上书店，通常都会有“书目信息”和“批发单”。一条“书目信息”面对不同的购买客户、可以存在多张“批发单”，反过来，一张“批发单”也可以批发多条书目，这就是多对多模式。中间的“批发单明细”表就是两者的关联表，具备独立的业务处理需求，是一个业务实体对象，因此它具备一些特有的属性，比如针对每一条明细记录而言的“累计退货次数”、“累计退货数量”、“累计结算次数”、“累计结算数量”；由于批发单明细在数据产生后已经打印出纸质清单提供给客户，因此在“批发单明细”表里对纸质清单中打印的书目信息属性作了冗余（逆标准化），这样在将来即使修改了“书目信息”表中的属性，也不会影响跟客户核对批发单明细，不会影响未来的财务结算业务。

关联表没有独立的业务处理需求。举例如下：比如用户与角色之间的关系，一般系统在做权限控制方面的程序时都会涉及到“系统用户表”和“系统角色表”。一个用户可以从属于多个角色，反过来一个角色里面也可以包含多个用户，两者也是典型的“多对多关系”。其中的关联表“用户角色关联表”在绝大多数情况

下都是仅仅用作表示用户与角色之间的关联关系，本身不具备独立的业务处理需求，所以也就没有什么特殊的属性。

### 5.1.3 使用上述四种模式的一般原则

#### 1. 什么时候用“主扩展模式”？

对象的个数不多；各个对象之间的属性有一定差别；各个对象的属性在数据库设计阶段能够完全确定；各个扩展对象有独立的、相对比较复杂的业务处理需求，此时用“主扩展模式”。将各个对象的共有属性抽取出来设计为“主表”，将各个对象的剩余属性分别设计为相应的“扩展表”，“主表”与各个“扩展表”分别建立一对一的关系。

#### 2. 什么时候用“主从模式”？

对象的个数较多且不固定；各个对象之间的属性几乎没有差异；对象的属性在数据库设计阶段能够完全确定；各个对象没有独立的业务处理需求，此时用“主从模式”。将各个对象设计为“从表”的记录，与“主表”对象建立一对多的关系。

#### 3. 什么时候用“名值模式”？

对象的个数极多；各个对象之间的属性有较大差异；对象属性在数据库设计阶段不能确定，或者在系统运行时有较大变更；各个对象没有相互独立的业务处理需求，此时用“名值模式”。

#### 4. 什么时候用“多对多模式”？

两个对象之间互为一对多关系，则使用“多对多模式”。

## 5.2 事务数据访问特性

事务（Transaction）是用户定义的数据库操作序列，这些操作要么都做，要么都不做。是一个不可分割的工作单位。

事务与应用程序是两个概念，一般来说，一个应用程序可以包含多个事务。事务的开始与结束可以由用户显式控制，如果用户没有显式定义事务，则由 DBMS 按缺省规定自动划分事务。

事务具有以下几个特性：

#### 1. 原子性（Atomicity）

事务中包括的所有操作要么都做，要么都不做。

#### 2. 一致性（Consistency）

事务执行的结果必须是使数据库从一个一致性的状态，变到另一个一致性的状态。

### 3. 隔离性（Isolation）

一个事务的执行不能被其他事物干扰。即一个事务内部的操作及使用的数据对其他并发事务是隔离的，并发执行的各个事务之间不能互相干扰。

### 4. 持久性（Durability）

一个事务一旦提交之后，它对数据库的影响必须是永久的。其他操作或故障不应该对其执行结果有任何影响。

事务的 ACID 特性对于数据库数据的正确、有效具有重要意义。但事务的特性可能遭破坏，主要有两种情况：

1. 多个事务并行运行时，不同事务的操作交叉进行。
2. 事务在运行过程中被强行停止。

利用数据库并发控制机制以及数据库恢复机制保证事务的特性不被破坏，从而保证数据库数据的正确有效。故事务是数据库恢复和并发控制的基本单位。

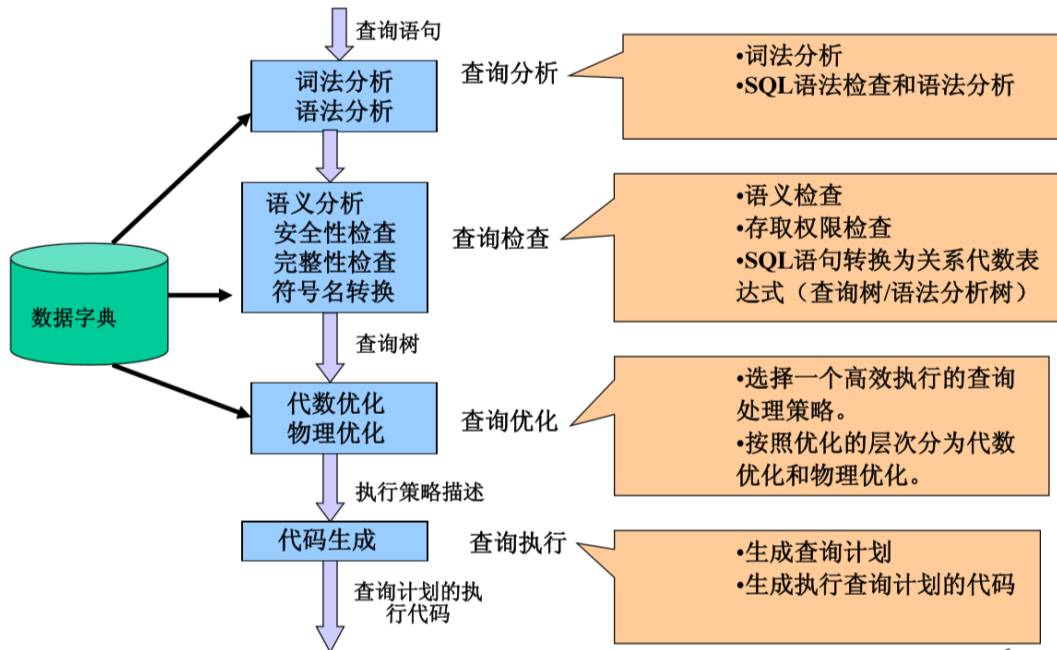
1. 原子性由恢复机制实现
2. 一致性是由事务的原子性保证的
3. 隔离性通过并发控制机构实现
4. 持久性通过恢复机制实现

在 SQL 中定义事务的语句有三条：

事务开始	BEGIN TRANSACTION	
事务结束	提交事务，正常结束	COMMIT
	撤销全部更新，回滚到事务开始时状态，非正常结束	ROLLBACK

### 5.2.1 查询事务

关系查询处理分为四个阶段：查询分析、查询检查、查询优化和查询执行。



#### • 学生查询宿舍信息

【事务内容】学生可以查询到自己宿舍的相关信息，例如宿舍号 Dno、宿舍电话 Dphone 以及可通过宿舍物品编号 Pno 查询到宿舍物品。需要连接表 student、dorm。

【操作频率】500 次/天

【性能要求】1s 内完成

#### • 学生查询宿舍物品报修信息

【事务内容】学生可以查询到自己宿舍的物品报修相关信息，例如报修的提交日期 Rsubmit、报修的原因 Rreason、报修的物品是什么 Pno 已经什么时候成功修好 Rsolve。需要连接表 student、repair、possession。

【操作频率】300 次/天

【性能要求】1s 内完成

#### • 学生查询宿舍卫生信息

【事务内容】学生可以查询到自己宿舍的卫生检查相关信息，例如什么时候被管理员检查了宿舍 ChTime 以及卫生评级情况 ChResult。需要连接表 student、check。

【操作频率】200 次/天

【性能要求】1s 内完成

- 学生查询宿舍水电信息

【事务内容】学生可以查询到自己宿舍的水电使用相关信息，例如已经用了多少钱 Fcost、宿舍还剩余多少水电费 Fleftfee 以及如果欠费之后续交了多少钱 Fctfee。需要连接表 student、fee。

【操作频率】500 次/天

【性能要求】1s 内完成

- 学生查询宿舍来访人员信息

【事务内容】学生可以查询到自己宿舍的来访人员相关信息，例如是谁来访宿舍 Vname、来访者性别 Vgender，以及来访者的来去时间 Vin 和 Vout。需要连接表 student、visitor。

【操作频率】200 次/天

【性能要求】1s 内完成

- 学生查询自己离校或退宿信息

【事务内容】学生可以查询到自己离校或者退宿的相关信息，例如离/返校时间 Lout/Lin，若退宿，可查到退宿时间 Qout 以及退宿原因 Qreason。需要连接表 student、quit、leave。

【操作频率】100 次/天

【性能要求】1s 内完成

- 宿舍管理员查询学生信息

【事务内容】宿舍管理员可以查询到寝室楼内所有学生的相关信息，例如学生学号 Sno、学生姓名 Sname、学生性别 Sgender、学生专业 Sdept、该学生所住的宿舍号 Dno 以及学生的入住时间 Scheckin。需要连接表 manager、student、dorm。

【操作频率】100 次/天

【性能要求】5s 内完成

- 宿舍管理员查询宿舍信息

【事务内容】宿舍管理员可以查询到寝室楼内所有宿舍的相关信息，例如宿舍号 Dno、宿舍电话 Dphone 以及可通过宿舍物品编号 Pno 查询到宿舍物品。需要连接表 manager、dorm。

【操作频率】500 次/天

【性能要求】1s 内完成

- **宿舍管理员查询宿舍物品报修信息**

【事务内容】宿舍管理员可以查询到寝室楼内所有宿舍的物品报修相关信息，例如报修的提交日期 Rsubmit、报修的原因 Rreason、报修的物品是什么 Pno 以及什么时候成功修好 Rsolve。需要连接表 manager、repair、possession。

【操作频率】1000 次/天

【性能要求】2s 内完成

- **宿舍管理员查询宿舍卫生信息**

【事务内容】宿舍管理员可以查询到寝室楼内所有宿舍的卫生检查相关信息，例如什么时候被管理员检查了宿舍 ChTime 以及卫生评级情况 ChResult。需要连接表 manager、check。

【操作频率】500 次/天

【性能要求】2s 内完成

- **宿舍管理员查询宿舍水电信息**

【事务内容】宿舍管理员可以查询到寝室楼内所有宿舍的水电使用相关信息，例如已经用了多少钱 Fcost、宿舍还剩余多少水电费 Fleftfee 以及如果欠费之后续交了多少钱 Fctfee。需要连接表 manager、fee。

【操作频率】1000 次/天

【性能要求】5s 内完成

- **宿舍管理员查询宿舍来访人员信息**

【事务内容】宿舍管理员可以查询到寝室楼内所有宿舍的来访人员相关信息，例如是谁来访宿舍 Vname、来访者性别 Vgender，以及来访者的来去时间 Vin 和 Vout。需要连接表 manager、visitor。

【操作频率】500 次/天

【性能要求】1s 内完成

- **宿舍管理员查询学生离校或退宿信息**

【事务内容】宿舍管理员可以查询到寝室楼内所有学生离校或者退宿的相关信息，例如离/返校时间 Lout/Lin，若退宿，可查到退宿时间 Qout 以及退宿原因 Qreason。需要连接表 manager、quit、leave。

【操作频率】500 次/天

【性能要求】1s 内完成

### 5.2.2 更新事务

- **学生注册、注销、修改自己的信息**

【事务内容】需要访问 student 表。如果是注册要往 student 表添加新的条目，其中一些条目可以为空；如果是注销或者修改已有资料，则需要通过索引找到指定 student 的条目，删除或者更新信息。如果删除 student 的话，那么这个 student 的相关信息都要级联删除。

【操作频率】100 次/天

【性能要求】1s 内完成

- **宿舍管理员注册、注销、修改宿舍的信息**

【事务内容】宿舍管理员 manager 需要访问 dorm、repair、check、fee、visitor 表。宿舍管理员对宿舍物品报修以及报修情况更新、宿舍水电费登记、宿舍卫生情况检查以及外来人员访问记录等具有增删改的权限。

【操作频率】500 次/天

【性能要求】1s 内完成

- **学生报修宿舍物品**

【事务内容】如果寝室内有物品损坏的情况，将会修改 possession 表。

【操作频率】100 次/天

【性能要求】1s 内完成

- **宿舍管理员创建新宿舍**

【事务内容】当前已有宿舍不够住进新同学，宿舍管理员将分配一间新的寝室，访问 dorm 表

【操作频率】10 次/月

【性能要求】5s 内完成



- 宿舍内人员变动

【事务内容】当已有宿舍内住进了新同学，或者有同学退宿时，管理员具有更新 dorm 权限。

【操作频率】50 次/月

【性能要求】1s 内完成

## 5.3 数据库索引设计

索引是指按表中某些关键属性或表达式建立元组的逻辑顺序，它是由一系列表元组的标识号组成的一个列表。使用索引可以快速访问表中的特定元组，被索引的表称为基表。索引并不改变表中元组的物理顺序，索引技术会将对于元组的逻辑排列保存在索引文件中。基表文件中的元组被修改或删除时，索引文件会自动更新以保证能够准确地找到新的数据。

索引是数据库设计和系统维护的一个关键部分，可以通过索引快速地查找数据和定位数据物理位置，从而大大减少查询执行时间。由于索引是冗余的数据结构，因此索引对保证正确性来说并不是必须的。但是，索引对事务的高效处理十分重要，既包括更新事务又包括查询。所以对完整性约束的有效实施也很重要。原则上，数据库系统可以自动决定创建何种索引。但是，由于索引的空间代价和索引对更新的影响，在维护何种索引上自动地做出正确选择并不容易。

假设有下面这样一个表：

```
CREATE TABLE student{  
    ID integer;  
    name varchar;  
}
```

需要大量使用类似下面这样的语句进行查询：

```
SELECT name FROM student WHERE id = 007;
```

通常，数据库系统需要一行一行地扫描整个 student 表以寻找所有匹配的元组、如果表 student 的规模很大，但是满足 WHERE 条件的只有少数几个（可能是 0 个或者 1 个），那么这种顺序扫描的性能就比较差了。如果让数据库系统在 id 属性上维护一个索引用于快速定位匹配的元组，那么数据库系统只需要在搜索树中查找少数的几层就可以找到匹配元组，这将大大提高数据查询的性能。类似的，在数据库中进行更新、删除操作时也需要先找到要操作的元组，利用索引同样可以提升这些操作的性能。

### 5.3.1 索引表

为数据库中个基本表建立的索引如下：

1. 由于基本表 **student** 的主键 **Sno** 经常在查询条件和连接操作的连接条件中出现，它的值唯一，考虑在 **Sno** 这个属性上建立唯一性索引。

如果索引声明为唯一索引，那么就不允许出现多个索引值相同的元组。唯一索引可以用于强迫索引属性数值的唯一性，或者是多个属性组合值的唯一性。唯一索引通过在创建索引命令中加上 **UNIQUE** 关键字来创建。一个多字段唯一索引认为只有两个元组的所有被索引属性都相同的时候才是相同的，这种重复元组才被拒绝。

2. 基本表 **student** 的属性 **Sname** 经常在查询条件中出现，且经常出现在相等的比较条件中，考虑在其之上建立聚簇索引。
3. 基本表 **possession** 的属性 **Pno** 经常在查询条件中出现，且与 **drom** 表的 **Dno** 组合值唯一，考虑在它们之上建立组合索引。
4. 由于 **repair**、**check**、**fee**、**visitor**、**leave**、**quit** 表的属性值经常发生变化，权衡系统为维护索引所付出的代价，暂考虑不在这几个表上建立索引。

## 5.4 完整性设计

### 5.4.1 对联系的定义

我设计的联系集如下。**checkin**、**manage**、**visit**、**repair**、**quit**、**leave**、**check**。

1. **checkin**: 关联学生和宿舍
2. **manage**: 关联宿舍管理员和宿舍
3. **visit**: 关联学生和来访人员
4. **repair**: 关联学生和宿舍财产
5. **quit**: 关联学生和宿舍
6. **leave**: 关联学生和宿舍
7. **check**: 关联宿舍管理员和宿舍

### 5.4.2 对约束的定义

1. **student** 表

属性名	约束
学号	主键，not null

姓名	not null
性别	‘female’ 或 ‘male’ , not null
专业	not null
宿舍号	外键, not null
入住时间	not null

## 2. dorm 表

属性名	约束
宿舍号	主键, not null
宿舍电话	not null
物品编号	外键, not null

## 3. repair 表

属性名	约束
宿舍号	外键, not null
提交日期	not null
解决日期	not null
报修原因	not null
物品编号	外键, not null

## 4. check 表

属性名	约束
宿舍号	外键, not null
检查时间	not null
检查结果	not null
物品编号	外键, not null

## 5. fee 表

属性名	约束
宿舍号	外键, not null
已用费用	not null

剩余费用	not null
续交费用	not null

#### 6. visitor 表

属性名	约束
姓名	not null
性别	‘female’ 或 ‘male’ , not null
进入时间	not null
离开时间	not null
学号	外键, not null

#### 7. leave 表

属性名	约束
学号	外键, not null
离校时间	not null
返校时间	not null

#### 8. quit 表

属性名	约束
学号	外键, not null
退宿时间	not null
退宿原因	not null

#### 9. manager 表

属性名	约束
管理员 ID	主键, not null
管理员密码	not null

#### 10. possession 表

属性名	约束
-----	----

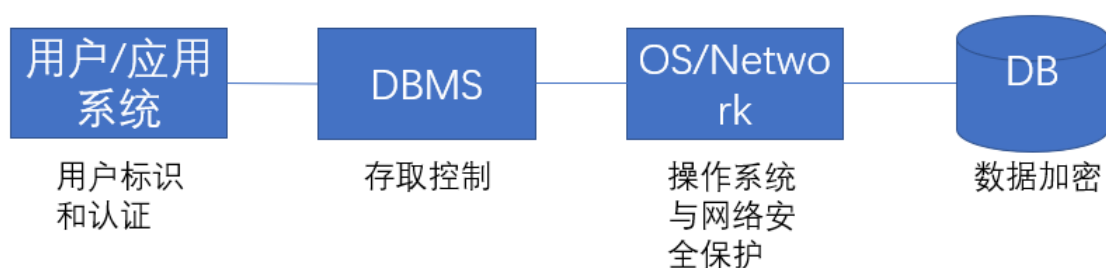
物品编号	主键, not null
物品名	not null
学号	外键, not null

## 5.5 安全性设计

数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改和破坏。它包括两方面的含义：

1. 向授权用户提供可靠的信息服务。
2. 拒绝非授权的对数据的存取访问请求，保证数据库管理下的数据的可用性、完整性和一致性，进而保护数据库所有者和使用者的合法权益。

下图是包含数据库系统的计算机系统安全模型：



**用户标识和认证**是系统提供的最外层安全保护措施。

1. 标识是指系统采用一定的方式标识其用户或应用程序的名字或身份。
2. 认证是指系统在用户或应用程序登录时判断其是否为合法的授权用户。

常用的方法是采用用户名和口令。

**存取控制**确保合法用户按照指定的权限使用 DBMS 和访问数据，而非法用户或不具有相关权限的用户则不能。

存取控制机制主要包括两个部分：

1. **用户权限定义**：将用户权限记录到数据库字典中，形成安全规则或授权规则。
2. **合法权限检查**：每当用户发出数据库操作请求后，DBMS 根据数据字典中的安全规则进行合法权限检查，决定是否接受用户的操作请求。

用户权限定义和合法权限检查机制一起组成了 DBMS 的安全子系统。

**自主存取控制**保证用户对于不同的数据对象拥有不同的存取权限，不同的用户对同一对象也有不同的权限，而且用户还可以将其拥有的权限转授给其他用户。

**强制存取控制**让每一个数据对象被标以一定的密级，每一个用户也被授予某一个级别的许可证。对于任一个对象，只有具有合法许可证的用户才可以存取。

在 SQL 中可以授予用户两类权限：

#### 1. 用户级权限

是数据库管理员为每个用户授予的特定权限。这种权限与整个数据库相关，与数据库中具体的关系无关。这种权限是对用户使用整个数据库的权限的限定。

#### 2. 关系级权限

是数据库管理员或数据库对象的拥有者为用户授予的与关系或视图有关的权限。这种权限是对用户使用关系和视图的权限的限定。

考虑到本项目开发设计的高校宿舍信息管理系统中存储有学生个人信息以及宿舍信息等敏感数据，一旦泄露将对学生与宿舍安全造成较大影响，因此需要采取必要的系统安全防护措施，有效保护系统内数据的安全性。系统的安全性需求主要是为了保证整个系统所有核心数据以及运行环境的安全。结合具体应用需求，设计了系统防火墙、内容过滤机制以及数据备份与恢复三大数据保护机制。

#### 1. 防火墙

为了便于学生用户对系统的访问，本系统直接接入互联网，这在给学生提供便利的同时，却也使系统直接面临着来自互联网的木马病毒、恶意攻击等一系列威胁，直接影响着系统的安全。为了保证系统的安全性，首先需要在系统与互联网连接的关键节点处设置防火墙，其作为系统的第一道防线，能够发挥系统屏障作用，有效识别拦截来自互联网的各类安全威胁，避免木马病毒等恶意文件进入系统，给系统造成损坏。除了拦截来自互联网的木马病毒外，防火墙还承担着用户鉴别的功能，能够对访问系统的用户的身份进行鉴别，一旦发现非法用户访问时，立即停止对该用户的响应，能够有效避免非法用户恶意访问系统可能造成的数据泄露或损毁。

#### 2. 内容过滤机制

尽管防火墙能够很大程度上减少系统所受的安全威胁，但仍不能完全保证系统不受到任何攻击，为了避免系统内部存储的敏感数据丢失，系统还在与互联网相连接的关键节点处设置了数据过滤屏障，并应用了内容过滤机制，利用预设的数据过滤准则与过滤策略，对进出节点的数据进行过滤，特别是对伪装成正常文件的数据进行排查，一方面避免伪装的木马病毒等恶意程序进入系统，另一方面避免系统内部敏感数据的泄露。

#### 3. 数据备份与数据恢复机制

考虑到本项目开发设计的高校宿舍管理系统对高校的正常运营有着重要的意义,为了进一步确保系统的安全稳定运行,系统还设置了数据备份与恢复机制,能够定期对系统内部的数据进行备份,一旦由于自然灾害等不可控因素造成系统损毁时,系统将自动唤醒灾难应急数据恢复机制,通过备份数据进行恢复,在最快的时间内恢复系统的正常稳定运行。

#### 5.5.1 用户管理

系统分为2级用户权限:宿舍管理员可以管理楼中的宿舍、宿舍内的学生等信息;学生可以直接查看宿舍成员的信息等。

#### 5.5.2 用户权限

宿舍管理员可以对宿舍房间信息和宿舍中的学生信息进行增删改查。学生可以对宿舍中的学生信息进行查询。

## 六、结论与展望

### 6.1 结论

上述分别从细节剖析出发,谈及系统相关性能需求分析,功能需求分析,业务需求分析,高校宿舍管理系统借助技术功能的替代要时刻发挥系统设计的校园整合管理优势。

由于学校宿舍的管理需求不断上升,从市场需求分析角度出发稳健地提高抓住使用者,不断提高竞争力扩大消费影响力以及利益需求,提出有效的寝室管理改进意见,分析寝室管理的业务内容。并且该系统有效地将校园数据信息系统和寝室管理结合在一起,运用数字化技术加以辅佐教学,从多角度出发提高校园的服务质量。

本文的设计理念是将一些计算机编程技术以及网络连接设备融入本文研究的高校宿舍管理系统方案当中使其有着一定的技术支撑。信息技术的存在是宿舍管理系统拥有校园信息共享,信息实时更近的功能保障。首先需要考虑的是系统收录的信息是否真实、准确,考察其是否和宿舍管理系统相吻合。宿舍管理系统的功能需求的研究是否符合现在社会的信息化建设以及校园文化发展。然而针对不少校园系统开发过程中发生技术排斥错误,信息不匹配等问题的出现,不少学校在技术开发方面做了一定的克服,为了校园教育文化环境的长期发展,学校寻求不同的方式更近,比如图片编辑以及美化需要具有吸引力,系统的稳定性以及系统管理的多方向性。

### 6.2 展望

#### 6.2.1 信息化校园管理启动了学生的便利生活

如今信息化建设以及无线网络的普及使信息化校园的拓展建设成为趋势,不少高校领导不再局限于原来传统的管理模式,网络平台的扩大是校园信息共享以及随时记录的技术服务。宽带无线 IP 网,网络连接设备以及计算机新编程技术实现了技术和管理相结合的创新思想,也就是信息化校园管理。信息化校园管理换一种说法就是可以有效借助计算机等相关技术运用于各大高校信息管理系统对大量的信息资源进行整合归类,信息化校园管理需要时间的实时更近,如同一部拓展化功能的移动设备,是师生,管理员校园活动以及线上,线下工作的有力的左右手,掀起了全球信息化建设的风波热潮。

#### 6.2.2 优化数据处理功能

数据处理功能是系统管理的一大特征,管理系统平台需要一定的访问量,每位用户使用过之后都会在平台都会留下痕迹,不管是校园共享信息的访问量系统保存的痕迹还是系统的后期维修。



### 6.2.3 系统性能兼容问题

系统的实际应用以校园宿舍信息管理为主的信息管理系统，为了能够更好的和学校其他信息管理系统相融合，宿舍管理系统需要对其性能操作兼容化，功能的普遍化。

## 七、系统实现

### 7.1 概述

该项目实现了学生信息、宿舍信息、管理员信息、考勤信息等的管理功能，学生、宿管、系统管理员登录后系统显示不同的界面。数据库采用的是 mysql，功能结构层次非常清晰，是数据库与 Java 学习的完整结合。下面来介绍如何运行。

1. 将项目导入到 eclipse 中；
2. 导入之后发现没有错误，如果有错误的话可能是 jre 需要配置一下。
3. tomcat 没有配置的时候，可能会出现 httpServlet 错误的提示，此时有两种方法可以解决，（1）配置 tomcat，运行项目的时候选择配置的 tomcat。还有一种是添加一个 httpServlet 的 jar 包，这个包我放在了项目根目录下，如果以前运行过 Java web 项目，那么上面那些问题都不会出现。
4. 接下来配置数据库信息，在 dorm.properties 文件中，配置自己的 mysql 用户名和密码；
5. 将数据库文件导入到 mysql 中，我用的是 navicat 工具；
6. 成功导入后就可以运行了，下面是运行结果

### 7.2 首页

系统登录页面如下，可以供系统管理、宿舍管理员和学生登录系统，需要输入用户名和密码。



图 7-1 系统主页

7.3 系统管理员

系统管理员主页如下，显示宿舍管理员管理、学生管理、宿舍楼管理、缺勤记录、修改密码和退出系统等功能栏。



图 7-2 系统管理员主页

系统管理员管理宿舍管理员，可以修改、添加、查看和删除宿舍管理员，以下为查看宿舍管理员信息，包括宿舍管理员的编号、姓名、性别、电话、所属宿舍楼、用户名等信息。

宿舍管理员管理						
<div>添加</div> <div>姓名 <input type="text"/> 搜索</div>						
编号	姓名	性别	电话	宿舍楼	用户名	操作
1	小白	男	123	4栋	manager1	<div>修改删除</div>
2	小张	男	123	2栋	manager2	<div>修改删除</div>
3	小李	女	123	1栋	manager3	<div>修改删除</div>
4	小陈	男	123	3栋	manager4	<div>修改删除</div>
5	小宋	男	123	无	manager5	<div>修改删除</div>
<div>首页 上一页 1 2 下一页 尾页</div>						

图 7-3 管理宿舍管理员

系统管理员添加宿舍管理员，需要填写用户名、密码、姓名、性别和联系电话等信息。

\*用户名：

\*密码：

\*重复密码：

\*姓名：

\*性别：

男

\*联系电话：

保存

返回

图 7-4 添加宿舍管理员

系统管理员管理学生，可以修改、添加、查看和删除学生，以下为查看学生信息，包括学生的学号、姓名、性别、电话、所属宿舍楼、寝室等信息。

学生管理

添加

全部宿舍楼

姓名

搜索

学号	姓名	性别	宿舍楼	寝室	电话	操作
001	测试1	男	1栋	111	123	<div>修改</div> <div>删除</div>
002	李四	男	2栋	120	32	<div>修改</div> <div>删除</div>
003	王五	男	3栋	201	2	<div>修改</div> <div>删除</div>
004	李进	女	4栋	220	1	<div>修改</div> <div>删除</div>
005	赵超	女	2栋	220	123	<div>修改</div> <div>删除</div>
006	王珂珂	女	2栋	111	111	<div>修改</div> <div>删除</div>
007	测试1	男	1栋	221	123	<div>修改</div> <div>删除</div>
008	测试3	男	4栋	123	123	<div>修改</div> <div>删除</div>
009	测试4	男	3栋	123	123	<div>修改</div> <div>删除</div>
010	小强	男	2栋	222	111	<div>修改</div> <div>删除</div>

图 7-5 管理学生

\*学号：

\*密码：

\*重复密码：

\*姓名：

\*性别：

请选择... ▾

\*宿舍楼：

1栋 ▾

\*寝室：

\*联系电话：

保存

返回

系统管理员管理宿舍楼，可以修改、添加、查看和删除宿舍楼，以下为查看宿舍楼信息，包括宿舍楼的编号、名称和简介等信息。

宿舍楼管理			
<a href="#">添加</a>		名称: <input type="text"/>	
编号	名称	简介	操作
1	1栋	这是一栋。。。	<a href="#">管理</a> <a href="#">修改</a> <a href="#">删除</a>
2	2栋	这是2栋	<a href="#">管理</a> <a href="#">修改</a> <a href="#">删除</a>
3	3栋	无	<a href="#">管理</a> <a href="#">修改</a> <a href="#">删除</a>
4	4栋	无	<a href="#">管理</a> <a href="#">修改</a> <a href="#">删除</a>
5	5栋	无	<a href="#">管理</a> <a href="#">修改</a> <a href="#">删除</a>
<div><a href="#">首页</a> <a href="#">上一页</a> <a href="#">1</a> <a href="#">2</a> <a href="#">下一页</a> <a href="#">尾页</a></div>			

系统管理员添加宿舍楼，需要填写宿舍楼名称和简介等信息。

\*名称：

简介：

保存

返回

53 / 60

系统管理员管理考勤记录，可以查看和删除考勤记录，以下为查看考勤记录信息，包括日期、学生学号、姓名、宿舍楼、寝室和备注等信息。

考勤记录						
日期	学号	姓名	宿舍楼	寝室	备注	操作
2014-01-01	002	李四	2栋	120	123	删除
2014-08-11	007	测试1	1栋	221	123	删除
2014-08-12	005	赵超	2栋	220	---	删除
2014-08-12	006	王珂珂	2栋	111	00	删除
2014-08-12	004	李进	4栋	220	---	删除
2014-08-12	004	李进	4栋	220	22	删除

图 7-9 管理考勤记录

7.4 宿舍管理员

宿舍管理员主页如下，显示学生查看、缺勤记录、修改密码和退出系统等功能栏。

宿舍管理系统

首页

学生查看

缺勤记录

修改密码

退出系统

欢迎您,宿舍管理员!

图 7-10 宿舍管理员主页

宿舍管理员管理学生，可以查看学生信息和给学生添加考勤记录，以下为查看学生信息，包括学生的学号、姓名、性别、电话、所属宿舍楼、寝室等信息。

学生管理						
学号	姓名	性别	宿舍楼	寝室	电话	操作
004	李进	女	4栋	220	1	添加缺勤记录
008	测试3	男	4栋	123	123	添加缺勤记录

图 7-11 管理学生

宿舍管理员考勤记录，可以添加、修改、查看和删除考勤记录，以下为查看考勤记录，包括日期、学生的学号、姓名、所属宿舍楼、寝室和备注等信息。

考勤记录						
添加	学号		姓名	宿舍楼	寝室	备注
日期	学号	姓名	宿舍楼	寝室	备注	操作
2014-08-12	004	李进	4栋	220	---	修改删除
2014-08-12	004	李进	4栋	220	22	修改删除

图 7-12 管理考勤记录

宿舍管理员添加考勤记录，需要填写学号、日期和备注等信息。

\*学号：

\*日期：

2017-12-26

\*备注：

保存

返回

图 7-13 添加考勤记录

7.5 学生

学生主页如下，显示缺勤记录、修改密码和退出系统等功能栏。

宿舍管理系统

首页

缺勤记录

修改密码

退出系统

欢迎您,同学!

图 7-14 学生主页

学生只能查看考勤记录，以下为查看考勤记录，包括日期、学生的学号、姓名、所属宿舍楼、寝室和备注等信息。

考勤记录					
日期	学号	姓名	宿舍楼	寝室	备注
2014-08-12	004	李进	4栋	220	...
2014-08-12	004	李进	4栋	220	22

图 7-15 查看考勤记录

## 7.6 系统测试

### 7.6.1 测试的目的与目标

在此系统进行初步实现之后，开始进行对系统进行测试，找出系统中存在的 Bug，通过测试，用提交的 Bug 报告来为以后软件的改进提供标准和参考，能够在以后的系统改进中找到依据。

测试后的软件各模块基本功能可以顺利进行，尽可能的提高软件的健壮性。

### 7.6.2 测试方法

从是否关心软件内部结构和具体实现的角度划分：黑盒测试和白盒测试；从是否执行程序的角度：静态测试和动态测试；从软件开发的过程按阶段划分有：单元测试、集成测试、确认测试、系统测试、验收测试、回归测试、Alpha 测试、Beta 测试；

(1) 单元测试又称模块测试，是针对软件设计的最小单位——程序模块（这里所说的程序模块在 Java 中一个模块就是一个方法），进行正确性检验的测试工作。其目的在于发现各模块内部可能存在的各种差错。单元测试需要从程序的内部结构出发设计测试用例。多个模块可以平行地独立进行单元测试。

(2) 集成测试（组装测试、联合测试），通常在单元测试的基础上，需要将所有模块按照设计要求组装成为系统。这时需要考虑的问题是：

- a. 在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失；
- b. 一个模块的功能是否会对另一个模块的功能产生不利的影响；
- c. 各个子功能组合起来，能否达到预期要求的父功能；
- d. 全局数据结构是否有问题；
- e. 单个模块的误差累积起来，是否会放大，从而达到不能接受的程度。

(3) 确认测试（Validation Testing），确认测试又称有效性测试。任务是验证软件的功能和性能及其它特性是否与用户的要求一致。对软件的功能和性能要求在软件需求规格说明书中已经明确规定。它包含的信息就是软件确认测试的基础。



(4) 系统测试 (System Testing)，是将通过确认测试的软件，作为整个基于计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其它系统元素结合在一起，在实际运行环境下，对计算机系统进行一系列的组装测试和确认测试。系统测试的目的在于通过与系统的需求定义作比较，发现软件与系统的定义不符合或与之矛盾的地方。

(5) 验收测试 (Acceptance Testing)，在通过了系统的有效性测试及软件配置审查之后，就应开始系统的验收测试。验收测试是以用户为主的测试。软件开发人员和 QA (质量保证) 人员也应参加。由用户参加设计测试用例，使用生产中的实际数据进行测试。在测试过程中，除了考虑软件的功能和性能外，还应应对软件的可移植性、兼容性、可维护性、错误的恢复功能等进行确认。

### 7.6.3 测试用例

由于功能模块较多，测试内容篇幅较长，所以在本论文中只介绍系统，下表是本系统会员登入的测试表，从测试的结果来看与期望结果完全相同。

登录系统测试用例

功能特性	用户登录验证				
测试目的	验证是否输入合法的信息				
测试数据	用户名称: 1111 密码: 1111 验证码: 正确的验证码				
测试内容	操作描述	数据	期望结果	实际结果	测试状态
1	输入用户名，按“登陆”按钮。	用户姓名: 1111, 密码为空	显示警告信息“用户名或密码误!”	显示警告信息“用户名或密码误!”	与期望结果相同
2	输入密码，按“登陆”按钮。	用户姓名为空，密码: 1111	显示警告信息“用户名或密码误!”	显示警告信息“用户名或密码误!”	与期望结果相同
3	输入用户名和密码，按“登陆”按钮。	用户姓名: 1, 密码: 1	显示警告信息“用户名或密码误!”	显示警告信息“用户名或密码误!”	与期望结果相同
4	输入用户名和密码，按“登陆”按钮。	用户名: 1111, 密码: 1111	正确登入到会员操作界面	正确登入到会员操作界面	与期望结果相同

#### 7.6.4 测试结论

把开始的代码写得越好，它出现的错误也就越少，你也就越能相信所做过的测试是彻底的。系统化测试以一种有序方式设法探测潜在的麻烦位置。同样，毛病最可能出现在边界，这可以通过手工的或者程序的方式检查。自动进行测试是最理想的，用得越多越好，因为机器不会犯错误、不会疲劳、不会用臆想某此实际无法工作的东西能行来欺骗自己。回归测试检查一个程序是否能产生与它们过去相同的输出。在做了小改变之后就测试是一种好技术，能帮助我们出现问题的范围局部化，因为新问题一般就出现在新代码里面。

测试和排错常常被说成是一个阶段，实际上它们根本不是同一件事。简单地说，排错是在你已经知道程序有问题时要做的事情。而测试则是在你在认为程序能工作的情况下，排错是在你已经知道程序有问题时要做的事情。而测试则是在你在认为程序能工作的情况下，为设法打败它而进行的一整套确定的系统化的试验。

Edsger Dijkstra 有一个非常有名的说法：测试能够说明程序中有错误，但却不能说明其中没有错误。他的希望是，程序可以通过某种构造过程正确地做出来，这样就不再会有错误了，因此测试也就不必要了。这确实是个美好生活的目标，但是，对今天的实际程序而言，这仍然还只是一个理想。所以应该集中精力讨论如何测试，如何才能更快地发现程序错误，如何才能使得工作更有成效、效率更高。

## 八、心得感想

设计开发基于 Jsp+Servlet+MySQL 的学生宿舍管理系统是集业务管理与软件技术为一身的工作，只有在理解业务的基础上，才能对业务操作流程进行提炼优化，并通过软件设计与开发技术实现。本文设计的学生宿舍信息管理系统界面友好、功能丰富、操作简单，能适应大部分环境下的学生宿舍管理系统所需要的工作。本系统在设计采用了 J2EE 框架，设计上实现了高内聚、低耦合的要求，在设计上做到了可扩展和维护。本文在论文撰写、系统开发时对现有学生宿舍管理系统做了深入的调研，并行性了需求分析和系统设计，最终实现了一个实用的学生宿舍管理系统。

通过开发《学生宿舍信息管理系统》，我较为全面的掌握了 JAVA 的基本知识和编程技巧，并在开发过程中我的 JAVA 开发能力得到了进一步的提高。比如：更进一步的深入了解 MVC 设计模式，软件三层框架；SQL 语言的使用。

在开发过程中我总结了一些经验：系统分析的好坏将决定着系统开发成功与否，一份好分析设计将是成功开发主要因素。我们在着手开发之前不要急于编程，先应有较长的时间去把分析做好，做好数据库设计工作，写出相关的开发文档等。然后再开始编写程序代码，这样做到每段代码心底都有数，有条不紊。当然也有些还需待继续深入地方如：控件的应用等。此外，我还觉得，我个人在这次设计中走了很多弯路。主要是因为平时很少接触软件开发工作，在应用方面缺乏经验，以后还需要更多的努力。

对我来说，这次设计的本身所产生的影响，还远远没有结束，我从本次课程设计中学到了许多课本上没有的知识。通过自己的学习和努力；通过老师的指导和教育，使我不仅仅在知识水平和解决实际问题的能力上有了很大的提高。还从思想的深处体会到，要把自己的所学变成现实时所面对的种种难题。

系统不免有错误和待改进之处，真诚欢迎各位师长、助教提出宝贵意见。

## 九、参考文献

- [1] AbrahamSilberschatz, Korth H , Sudarshan S , et al. 数据库系统概念 [M]. 高等教育出版社, 2006.
- [2] 曾玉红.基于 B/S 高校宿舍管理系统的设计与实现[J].电脑知识与技术,2018,14(36):60-62.
- [3] 万德生. 高校宿舍管理系统的设计与实现[D].吉林大学,2016.
- [4] 乌日娜. 集宁师范学院智能宿舍管理系统的设计与实现[D].大连理工大学,2018.
- [5] 马海英. 四川化工职业技术学院学生宿舍管理系统的设计与实现[D].电子科技大学,2018.
- [6] Silberschatz A , Korth H F , Sudarshan S . Database System Concepts, 6/e[J]. 2002.