江苏海洋大学计算机工程学院

课程设计报告

**设计名称：**  数据库课程设计

**选题名称：** 某高校的科研项目管理系统

**姓 名：** 陈康 **学 号：** 2018140580

**专业班级：** Z软件162

**系 （院）： 应用技术学院**

**设计时间：** 2019.10.01~2019.11.30

**设计地点：** SD309

|  |
| --- |
| 指导教师评语：  需求分析（较、欠）充分，  概念设计（较、不）完整，  逻辑设计（较、不）正确，  代码（较、不）正确，  系统运行（较、不）正常，  界面（较、不）美观，  报告格式（较、不）正确。  签名：  2019 年 11 月 30 日 |

|  |
| --- |
| 1．课程设计目的 |
| （1）培养学生运用所学课程《数据库系统原理》的理论知识和技能，深入理解《数据库系统原理》课程相关的理论知识，学会分析实际问题的能力。  （2）培养学生掌握用《数据库系统原理》的知识设计计算机应用课题的思想和方法。  （3）培养学生调查研究、查阅技术文献、资料、手册以及编写技术文献的能力。  （4）通过课程设计的训练，要求学生在教师的指导下，独立完成大作业。 |
| 2．课程设计任务与要求： |
| **任务：**   1. 通过调查研究和运用Internet，收集和调查有关资料、最新技术信息。 2. 基本掌握撰写小论文的基本步骤和写作方法。 3. 根据课题的要求基本理解和掌握数据流图（DFD）和数据字典（DD）的设计方法。 4. 根据课题的要求基本理解和掌握E-R图的设计方法和关系模式的转换。 5. 根据ER图生成数据库表。 6. 数据库完整性、安全性保证措施 7. 数据库实施维护计划   **基本要求：**   1. 每个学生从给定题目中任选一个作为课程设计题目，调查分析一个具体的或模拟的实例。同一个班中的同学不允许选择相同的题目。选好题目后发给班长或学习汇总，产生选题冲突时，由班长/委员协调解决； 2. 描述该实例的业务信息和管理工作的要求； 3. 建立数据字典； 4. 列出实体、联系； 5. 指出实体和联系的属性； 6. 画出E-R图； 7. 将E-R图转换成关系模式，并注明主码和外码； 8. 创建数据库； 9. 根据题目的要求写查询、存储过程、触发器等。 |

|  |
| --- |
| 3．课程设计说明书 |
| * 1. 需求分析      1. **问题描述**   试设计一个某高校的科研项目管理系统，下列各项为对该系统数据库的基本要求。   1. 实现部门、职务、职称等基本信息的管理； 2. 实现教师信息的管理； 3. 实现科研项目的申报、审批管理； 4. 实现科研项目的验收管理； 5. 创建默认，并绑定到科研项目的验收标志，使其默认值为“未验收”； 6. 创建触发器，验收项目时自动修改项目的验收标志为“验收通过”； 7. 创建存储过程统计个院系科研项目的申报和完成数量； 8. 建立数据库相关表之间的参照完整性约束；    * 1. **数据需求**   以下为我们需要的各种数据信息的罗列。   1. 教师：姓名、工号、性别、所在部门、职能、职称、年龄、联系方式等。 2. 申报项目：编号、预算、项目状态、名称、开始时间、借书时间、审核状态、审核备注、审核人、附件、项目介绍等。 3. 登录信息：工号、密码等。 4. 权限信息：审核权限、验收权限等。    * 1. **功能需求**   根据对问题的分析，我们大致将系统所需要的功能划分为四个模块，分别是登录模块、项目申请模块、项目审核模块、项目验收模块。以下是对各个模块所需功能的概述。   1. 登录模块：高校科研项目管理系统是针对高校所有教职人员提供服务的系统。为了保障数据的安全性，所以需要输入正确的工号和密码才能进行使用。 2. 项目申请模块：高校科研项目管理系统的主要作用是给教职人员的科研项目提供BS模式的管理功能。该模块的功能是允许需要提交项目申请的教职人员可以使用该系统进行项目的申请。并查看项目的各项进度。 3. 项目审核模块：对已提交申请但未审核的项目进行审核。允许查看该项目的申请信息并选择审核通过或审核不通过。 4. 项目验收模块：对审核通过的项目进行验收。运行选择验收通过或验收不通过。    * 1. **安全性及完整性需求** 5. 安全性需求分析  * 需要输入工号与密码才能登陆本系统； * 输入的登陆密码需要以“\*”显示； * 禁止非法数据写入； * 对于各种错误可以做到提醒或纠正。  1. 完整性需求  * 数据库设计时要使关系的属性不可再分，满足原子性； * 约束表格中部分属性的内容； * 为数据的新增、修改或删除设定对应的触发器； * 防止数据库查询发生 SQL 注入的攻击。   + 1. **数据流图**     图3 - 1高校科研项目管理系统顶层数据流图    图3 - 2高校科研项目管理系统中层数据流图   * + 1. **数据字典**  1. 数据项   数据项名称：部门编号  别称：部门  简述：教师所在的部门  类型：int  数据项名称：工号  别称：教师工号  简述：教师的工号，教师的唯一标识  类型：char(10)  数据项名称：职能编号  别称：职能  简述：教师的职能  类型：int  数据项名称：职称编号  别称：职称  简述：教师的职称  类型：int  数据项名称：审核状态  别称：审核  简述：项目的审核状态  类型：varchar  数据项名称：项目编号  别称：项目号  简述：项目的编号，项目的唯一标识  类型：varchar  数据项名称：权限  别称：用户权限  简述：用户的操作权限  类型：int  数据项名称：验收状态  别称：验收  简述：项目的验收状态  类型：varchar   1. 数据结构  |  |  |  | | --- | --- | --- | | 数据结构名称：用户  简述：登录系统时的用户的基本信息  数据结构组成：用户名+密码 |  | 数据结构名称：项目  简述：项目的基本信息  数据结构组成：项目名+项目编号+审核状态+验收状态+经费+开始时间+借书时间 |  * 1. 系统设计      1. **概念设计**  1. 局部ER图     图3 - 3用户信息表ER图    图3 -4部门ER图    图3 -5 职能ER图    图3 -6项目信息表ER图   1. 全局ER图     图3 -7全局ER图   * + 1. **逻辑设计**   以下为本系统数据库的数据关系模型，带下划线的属性为主码。  teacher（编号，密码，工号，部门，职称，职能，联系方式，年龄，性别）  project（编号，名称，用户号，开始时间，结束时间，预算，项目号，简介，项目状态，审核状态，审核原因，审核人）  position（编号，名称）  title（编号，名称）  department（编号，名称）  其中teacher对应的基本关系，project对应项目的基本关系，position对应职能的基本关系，department对应部门的基本关系，title对应职称的基本关系。   * + 1. **数据库设计**  1. 物理结构图     图3 -8据库物理结构图  (2) 表结构  表3 - 1学生信息表   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 字段名 | 数据类型 | 约束 | 是否可为空 | 备注 | | 编号 | Int | 主键 | 否 | 主键 | | 姓名 | Varchar |  | 否 | 用户的姓名 | | 性别 | Int |  | 是 | 用户的性别 | | 密码 | Varchar |  | 否 | 用户的密码 | | 工号 | Varchar | 唯一 | 否 | 用户的工号 | | 部门编号 | Int | 外键 | 否 | 用户的部门编号 | | 职能编号 | Int | 外键 | 否 | 用户的职能编号 | | 职称编号 | Int | 外键 | 否 | 用户的职称编号 | | 联系方式 | Varchar |  | 是 | 用户的联系方式 |   表3 – 2项目信息表   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 字段名 | 数据类型 | 约束 | 是否可为空 | 备注 | | 编号 | Int | 主键 | 否 | 主键 | | 名称 | Varchar |  | 否 | 项目名 | | 项目号 | Varchar | 唯一 | 否 | 项目的编号 | | 用户编号 | Int | 外键 | 否 | 项目所属的用户 | | 开始时间 | Date |  | 是 | 开始时间 | | 借书时间 | Date |  | 是 | 借书时间 | | 预算 | Double |  | 是 | 项目预算 | | 简介 | Clob |  | 是 | 项目的简介 | | 状态 | Varchar |  | 是 | 项目的状态 | | 审核状态 | Varchar |  | 是 | 项目的审核状态 | | 审核人 | Int | 外键 | 是 | 项目的审核人id | | 审核原因 | Varchar |  | 是 | 审核人审核时填写的备注 |   表3 -3职能表   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 字段名 | 数据类型 | 约束 | 是否可为空 | 备注 | | 编号 | Int | 主键 | 否 | 主键 | | 名称 | Varchar |  | 否 | 职能名 |   表3 – 4部门表   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 字段名 | 数据类型 | 约束 | 是否可为空 | 备注 | | 编号 | Int | 主键 | 否 | 主键 | | 名称 | Varchar |  | 否 | 部门名 |   表3 – 5职称表   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 字段名 | 数据类型 | 约束 | 是否可为空 | 备注 | | 编号 | Int | 主键 | 否 | 主键 | | 名称 | Varchar |  | 否 | 职称表 |      * + 1. **系统环境** * 操作系统Ubuntu * DBMS：MySQL * 开发语言：Java * 开发平台：IDEA * 数据库管理工具：Navicat   1. 系统实现      1. **开发目标**   随着高校学生的数量的增加，对于学生学籍的管理也越来越困难了，除了传统的纸质信息管理，现在最常用的就是通过计算机进行电子化的信息管理，这样不仅仅大大减少了学籍管理人员的负担，而且还提高了信息的安全程度。综上，我们需要设计一款高校学籍管理系统来帮助高校更好地来完成学籍管理工作，从而减轻学籍管理人员的负担，并使得学籍的管理更加规范化和系统化。   * + 1. **系统需求**  1. 登录需求   打开系统时显示登陆界面，只有输入正确的用户名和对应的密码方能成功登录，输入用户名或密码错误时会有对应的提示信息出现。   1. 学生信息管理需求   可以查询学生的信息，包括学生的学号、姓名、性别、民族、出生日期和班级编号信息；可以添加新的学生信息，并且添加信息后要保证学号为学生信息的唯一标识；可以在输入学号后删除对应的学生信息；可以在输入学号后修改对应的学生信息；可以在输入学号后搜索并输出对应学生信息及其所在的班级信息；对于输入错误的情况可以进行提示，并且操作成功后也可以进行提示。   1. 班级信息管理需求   可以查询班级的信息，包括班级的班级编号、院系、专业、班号、班主任和班级人数信息；可以添加新的班级信息，并且添加信息后要保证班级编号为班级信息的唯一标识；可以在输入班级编号后删除对应的班级信息；可以在输入班级编号后显示对应班级的男生和女生人数；对于输入错误的情况可以进行提示，并且操作成功后也可以进行提示。   1. 课程信息管理需求   可以查询课程的信息，包括课程的课程号、课程名和课程类型信息；可以添加新的课程信息，并且添加信息后要保证课程号为课程信息的唯一标识；可以在输入课程号后删除对应的课程信息；对于输入错误的情况可以进行提示，并且操作成功后也可以进行提示。   1. 上课信息管理需求   可以查询上课的信息，包括上课的班级编号、课程号和上课教师信息；可以添加新的上课信息，并且添加信息后要保证（班级编号，课程号）为上课信息的唯一标识；可以在输入班级编号和课程号后删除对应的上课信息；对于输入错误的情况可以进行提示，并且操作成功后也可以进行提示。   1. 学生成绩管理需求   可以查询学生成绩的信息，包括成绩的学号、课程号和成绩信息；可以添加新的成绩信息，并且添加信息后要保证（学号，课程号）为成绩信息的唯一标识；可以在输入学号和课程号后删除对应的成绩信息；可以在输入学号和课程号后修改对应的学生成绩信息；对于输入错误的情况可以进行提示，并且操作成功后也可以进行提示。   1. 学生奖惩管理需求   可以查询学生奖惩的信息，包括奖惩的奖惩编号、学生学号、奖惩类型、奖惩内容和创建时间信息；可以添加新的奖惩信息，并且添加信息后要保证奖惩编号为奖惩信息的唯一标识；可以在输入奖惩编号后删除对应的奖惩信息；对于输入错误的情况可以进行提示，并且操作成功后也可以进行提示。   * + 1. **系统功能结构**     图3 - 21系统功能结构图   * + 1. **系统功能实现**  1. 登录功能     图3 - 22登录功能流程图   |  | | --- | | **关键代码**：user\_name = var\_user\_name.get()  user\_password = var\_user\_password.get()  sql1 = **"select** *\** **from login where 用户名 = '%s';"** % (user\_name)  mycursor.execute(sql1)  row = mycursor.fetchone() |  1. 学生信息管理功能     图3 - 23学生信息管理功能流程图   |  | | --- | | **关键代码：**  查询：cursor.execute(**"select** *\** **from student order by 学号 asc;"**)  results = cursor.fetchall()  添加：sql = **"insert into student values('%s','%s','%s','%s','%s-%s-%s',** \  **"'%s')"**% (g1, g2, g3, g4, g5\_1, g5\_2, g5\_3, g6)  **try:**  cursor.execute(sql)  db.commit()  **except:**  db.rollback()  删除：sql = **"delete from student where 学号='%s';"** % (g)  修改：sql = **"update student set 学号='%s', 姓名='%s', 性别='%s', 民族='%s',** \  **"出生日期='%s', 班级编号='%s' where 学号='%s'"** % \  (g1, g2, g3, g4, g5, g6, s.get())  搜索：sql = **"select student.姓名,student.性别, student.民族, "** \  **" student.班级编号, class.院系,class.专业,class.班号 "** \  **"from student,class where student.班级编号=class.班级编号 and"** \  **"student.学号='%s'"** % \(g)  cursor.execute(sql) |  1. 班级信息管理功能     图3-24班级信息管理功能流程图   |  | | --- | | **关键代码：**  查询：cursor.execute(**"select** *\** **from class order by 班级编号 asc;"**)  results = cursor.fetchall()  添加：sql = **"insert into class values('%s','%s','%s','%s','%s','0')"** % \  (g1, g2, g3, g4, g5)  **try:**  cursor.execute(sql)  db.commit()  **except:**  db.rollback()  删除：sql = **"delete from class where 班级编号='%s';"** % (g)  查询班级人数：cursor.execute(**'call sex\_num\_b(%s,@num\_b);'**% (g))  cursor.execute(**'select @num\_b;'**)  db.commit()  row1 = cursor.fetchone()  cursor.execute(**'call sex\_num\_g(%s,@num\_g);'**% (g))  cursor.execute(**'select @num\_g;'**)  db.commit()  row2 = cursor.fetchone() |  1. 课程信息管理功能     图3-25课程信息管理功能流程图   |  | | --- | | **关键代码：**  查询：cursor.execute(**"select** *\** **from course order by 课程号 asc;"**)  results = cursor.fetchall()  添加：sql = **"insert into course values('%s','%s','%s')"** % (g1, g2, g3)  **try:**  cursor.execute(sql)  db.commit()  **except:**  db.rollback()  db.close()  删除：sql = **"delete from course where 课程号='%s';"** % (g)  **try:**  **if** tkinter.messagebox.askquestion('提示', '您确认删除该课程信息吗？') == **'yes'**:  cursor.execute(sql)  db.commit()  **except:**  db.rollback()  db.close() |  1. 上课信息管理功能     图3 -26上课信息管理功能流程图   |  | | --- | | **关键代码：**  查询：cursor.execute(**"select** *\** **from allocate order by 课程号 asc,** **\**  **班级编号 asc,上课教师 asc;"**)  results = **cursor**.fetchall()  添加：sql = **"insert into allocate values('%s','%s','%s')"** % (g1, g2, g3)  **try:**  cursor.execute(sql)  db.commit()  **except:**  db.rollback()  删除：sql = **"delete from allocate where 班级编号='%s' and 课程号='%s';"** % \  (g1, g2)  **try:**  **if** tkinter.messagebox.askquestion('提示', '您确认删除该上课信息吗？') == **'yes'**:  cursor.execute(sql)  db.commit()  **except:**  db.rollback() |  1. 学生成绩管理功能     图3 -27成绩信息管理功能流程图   |  | | --- | | **关键代码：**  查询：cursor.execute(**"select** *\** **from score order by 课程号 asc,成绩 desc;"**)  results = cursor.fetchall()  添加：sql = **"insert into score values('%s','%s','%s')"** % (g1, g2, g3)  **try:**  cursor.execute(sql)  db.commit()  **except:**  db.rollback()  删除：sql = **"delete from score where 学号='%s' and 课程号='%s';"** % (g1, g2)  **try:**  cursor.execute(sql)  db.commit()  **except:**  db.rollback()  修改：sql = **"update score set 成绩='%s' where 学号='%s' and 课程号='%s';"** % \  (g1, s.get(), s\_.get())  cursor.execute(sql)  db.commit() |  1. 学生奖惩管理功能     图3 -28奖惩信息管理功能流程图   |  | | --- | | **关键代码：**  查询：cursor.execute(**"select** *\** **from reward\_or\_punishment order by 奖惩编号 asc;"**)  results = cursor.fetchall()  添加：sql = **"insert into reward\_or\_punishment** \  **values('%s','%s','%s','%s',curdate())"** % (g1, g2, g3, g4)  **try**:  cursor.execute(sql)  db.commit()  **except**:  db.rollback()  删除：sql = **"delete from reward\_or\_punishment where 奖惩编号='%s';"** % (g)  **try**:  **if** tkinter.messagebox.askquestion(**'提示'**, **'您确认删除该奖惩信息吗？'**) == **'yes'**:  cursor.execute(sql)  db.commit()  **except**:  db.rollback() |  1. 触发器  |  | | --- | | **#当添加学生时，为对应的班级人数增加一人**  create trigger student\_insert after insert on student for each row  begin  declare a int;  set a=(select 班级人数 from class where 班级编号=new.班级编号);  update class set 班级人数=a+1 where 班级编号=new.班级编号;  end  **#当删除学生时，为对应的班级人数减少一人**  create trigger student\_delete after delete on student for each row  begin  declare a int;  set a=(select 班级人数 from class where 班级编号=old.班级编号);  update class set 班级人数=a-1 where 班级编号=old.班级编号;  end  **#当修改学生时，为对应的原班级人数减少一人，并且为对应修改后的班级增加一人**  create trigger student\_update after update on student for each row  begin  declare a int;  set a=(select 班级人数 from class where 班级编号=old.班级编号);  update class set 班级人数=a-1 where 班级编号=old.班级编号;  set a=(select 班级人数 from class where 班级编号=new.班级编号);  update class set 班级人数=a+1 where 班级编号=new.班级编号;  end |  1. 存储过程  |  | | --- | | **#统计对应班级男生人数**  create procedure sex\_num\_b(in clas char(20) character set utf8,out num\_b int)  select count(\*) into num\_b from student where student.班级编号=clas and student.性别='男';  **#统计对应班级女生人数**  create procedure sex\_num\_g(in clas char(20) character set utf8,out num\_g int)  select count(\*) into num\_g from student where student.班级编号=clas and student.性别='女'; |  * 1. 系统评价      1. **主要优点** * 操作简便，易于上手； * 无需指导，自带提示； * 基本操作功能较为齐全。   + 1. **需要改进** * 在时间足够的情况下，本系统的功能应该可以更加齐全，特别是在查看和搜索方面应该更加灵活； * 本系统的界面设计还有待完善； * 本系统应该考虑到更多人性化的操作设计。 |

|  |
| --- |
| 4．课程设计成果 |
| * 1. 登录界面   如下4.1组图为本高校学籍管理系统的登录界面设计情况。    图4- 1登录界面设计   |  |  | | --- | --- | | 图4- 2用户名输入错误的提示信息 | 图4 - 3密码输入错误的提示信息 | | 图4 - 4用户名与密码输入正确后的提示信息 | 图4 - 5单击退出按钮提示 |  * 1. 学生信息管理界面   如下4.2组图为本高校学籍管理系统的学生信息管理界面设计情况（界面过大，故只保留有数据部分界面截图）。    图4 -6查询学生信息界面    图4 -7添加学生信息界面    图4 -8删除学生信息界面    图4 -9修改学生信息前的界面    图4 -10修改学生信息中的界面    图4 -11搜索学生信息成功后的界面   * 1. 班级信息管理界面   如下4.3组图为本高校学籍管理系统的班级信息管理界面设计情况。    图4 - 12查询班级信息界面    图4 -13添加班级信息界面    图4 -14删除班级信息界面    图4 -15查询班级男女人数界面    图4 -16查询班级男女人数结果显示   * 1. 课程信息管理界面   如下4.4组图为本高校学籍管理系统的课程信息管理界面设计情况。    图4 - 17查询课程信息界面    图4 -18添加课程信息界面    图4 -19删除课程信息界面   * 1. 上课信息管理界面   如下4.5组图为本高校学籍管理系统的上课信息管理界面设计情况。    图4 - 20查询上课信息界面    图4 - 21添加上课信息界面    图4 -22删除上课信息界面   * 1. 成绩信息管理界面   如下4.6组图为本高校学籍管理系统的成绩信息管理界面设计情况。    图4 -23查询成绩信息界面    图4 - 24插入成绩信息界面    图4 -25删除成绩信息界面    图4 -26修改成绩信息前的界面    图4 -27修改成绩信息中的界面   * 1. 奖惩信息管理界面   如下4.7组图为本高校学籍管理系统的奖惩信息管理界面设计情况。    图4 -28查询奖惩信息界面    图4 - 29插入奖惩信息界面    图4 - 30删除奖惩信息界面   * 1. 提示信息弹窗   如下4.8组图为本高校学籍管理系统的提示信息弹窗设计情况（因为几乎每一个带输入的界面都有提示信息弹窗，故只列举部分作为代表）。   |  |  | | --- | --- | | 图4-3 1文本框输入信息不存在的情况下提交后的提示信息 | 图4 -3 2文本框输入内容不符合规范的情况下提交后的提示信息 |  |  |  | | --- | --- | | 图4 - 33修改信息提交时的确认提示信息 | 图4 -34删除信息提交时的确认信息 | | 图4 -3 5操作成功后的提示信息 | 图4 -3 6登录本系统后想要退出时的提示信息 |     图4 -3 7删除信息成功后的提示信息  前面已经展示过的截图，包括其上数据都是经过对本次设计的高校学籍管理系统进行操作而得来的。  由于本系统的功能模块较多，所以就不将所有操作过程的截图都依次展示出来了，所有功能划分大类的话分别是查询功能、插入功能、删除功能、修改功能以及搜索功能等等，同大类功能界面操作比较相似，不再一一赘述。 |

|  |
| --- |
| **5.课程设计心得** |
| 本次课程设计完成了对高校学籍管理系统的设计，最终实现了对于学籍管理系统所需的基本功能的实现。这是个系统学习的过程，让我对所学的数据库知识加以巩固和加强，知道了如何去综合多方面的知识来设计数据库和进行界面的设计，我的数据库实际设计能力也大大提升。  此次数据库课程设计耗时近两周，参考了其他人的代码，他的能力很强并且用了我们从未学习过的语言来写，困难的不是看到程序的运行结果，困难的是要彻底弄明白使用的另外的语言的意义何在与其他我们学习过的语言有何不同。此次课程设计让我收获最大的是我理解并掌握了Python语言。此次的课程设计是基于Python语言以及MySQL数据库来进行设计开发的，使用的编程工具为PyCharm，而建立数据库方面起初并没有安装可视化界面，所以只能用命令控制符来进行MySQL数据库的操作，而到后期由于命令控制符查看数据库实在是不怎么方便，于是安装了Navicat软件来通过它的可视化界面来测试其存储数据的变化情况。  对于自己并不熟悉的语言与编程工具就要求我要花大量的时间去学习并掌握它，当然有了已经做好的代码作参考大大降低了难度，但是依然需要深入的理解，许多细节的地方依然要查找许多参考资料才能得出结果，这种一边参考一边学习的方式确实是另一番体验，此次实验为我以后自学其他的语言做好铺垫，以便能够快速接受新的技术。  关于本次系统的界面设计，具体来说是基于Python的tkinter库，它是一个Python自带的可视化控件库，其中包含有15种基本控件，在这次系统设计中并没有全部使用。Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言。Python 的设计具有很强的可读性，相比其他语言经常使用英文关键字，其他语言的一些标点符号，它具有比其他语言更有特色语法结构。所以在原有的代码的基础上我又添加了部分功能以实现高校学籍管理系统所需的功能。同时，完成课设过程中遇到了一些问题，比如对数据库的参照完整性掌握不透，导致各个数据表之间的关系初期较为凌乱，以及系统实际使用效率低下，缺少功能、功能冗余等问题，仍需要改善。  通过两个星期的数据课程设计，我从中受益匪浅，并且对数据库这一门课程有了更深一步的认识。在课程设计过程中，我可以把这个学期所学的理论知识和实践联系起来，在所要开发的程序中渐渐会贯通。虽然我对这些知识还运用得还不是很熟练，但是相信在现在和今后的学习中会得到更加深刻的掌握。课程设计过程中我们一边设计一边探索，发现理论和实践要充分地结合，是需要扎实的基本功的，这就表明学好基础知识是理论付诸实践的前提。在课程设计中我学到了很多，希望在以后我能充分利用课程设计的机会充实自己。 |