**数据库系统课程设计报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 姓名 | | 马欣萌 | | 班级 | | 计科181 |
| 学号 | | 181002424 | | 指导教师 | | 崔晓晖 |
| 组名 | | 想穿越到过年 | | | | |
| 团队其他成员（可自行增加行数，第一为组长） | | | | | | |
| 姓名 | | | 学号 | | 班级 | |
| 马欣萌 | | | 181002424 | | 计科181 | |
| 宫珍妮 | | | 181002427 | | 计创18 | |
| 王昕颖 | | | 181002329 | | 计科182 | |
| 1、个人和团队 | | | | | | |
| **课程设计概述** | （1）研究生助教工作的主要要求  研究生需从事至少一门本科课程或一门研究生课程助教工作。  学院根据选课人数等情况形成助教课程列表，供研究生助教选择和申报。该列表优先考虑学时量较大、选课人数较多课程。  助教采取研究生申报，授课教师选择的方法进行选课。每门课程只能指定一名助教，每名研究生可填报2项志愿。对于多轮选择后没有获得助教课程的研究生，由学院统筹确定助教课程。  授课教师负责根据助教工作情况填写《信息学院研究生助教工作评定表》（简称评定表）。评定结果合格方可申请毕业答辩，评定结果不合格需重新申请担任助教工作。研究生申请毕业答辩前，提交评定表到学院备案。  （2）研究生学术交流情况的主要要求  研究生需参加由导师认定的至少2次国内和国际高水平学术交流活动。  为保障学术交流的质量，研究生需在参与学术交流后，向导师及时汇报交流情况，留存参会证明资料。  申请毕业答辩前，选择2次国内或国际学术交流情况，填写《研究生学术交流统计表》（简称统计表）。统计表由导师和学科负责人签字后，上交学院备案。  （3）研究生参与项目情况的主要要求  研究生需完成指导教师指定的科研项目（6万元经费及以上的开发工作量），科研经费须经指导教师和项目负责人共同签字认定，不允许重复认定。  完成任指定科研项目后，填写《研究生参与项目认定表》。 | | | | | |
| **个人任务情况** | 本人在该课程设计中主要负责研究生助教系统的数据库的相关设计、开发和实现，全局ER图的设计和总体数据库的实现，并且负责了数据库的持久层框架设计和测试工作。  在助教系统的数据库的设计工作重点是对数据进行抽取和对如何实现数据库内容进行构思和设计。  在实现全局ER图的设计的重点是在于如何将不同人员设计的ER图寻找接口来进行实现。  在实现总体数据库设计的重点内容是了解如何进行数据库中各个表之间的协同，如何合理的设计外键。 | | | | | |
| **团队协作情况** | 由于在本次课程设计中担当组长，所以与两位组员都进行了协作。和不同组员的协作内容不同。  与宫珍妮同学的协作主要体现在全局ER图的整合和对于整体数据库内容的修改工作。  与王昕颖同学的协作主要体现在全局ER图的整合和数据库测试数据的设计和填充时进行沟通来设计出更加符合数据库设计的内容。  在全局ER图构建时和两位组员的合作建立在共同使用成员的接口上，在每一位同学设计完成自己的子系统的ER图后抽取相同部分作为接口来进行组合。 | | | | | |
| 2、问题分析 | | | | | | |
| **业务需求分析** | 本人主要负责的是研究生助教工作的主要要求，要求如下：  研究生需从事至少一门本科课程或一门研究生课程助教工作。  学院根据选课人数等情况形成助教课程列表，供研究生助教选择和申报。该列表优先考虑学时量较大、选课人数较多课程。  助教采取研究生申报，授课教师选择的方法进行选课。每门课程只能指定一名助教，每名研究生可填报2项志愿。对于多轮选择后没有获得助教课程的研究生，由学院统筹确定助教课程。  授课教师负责根据助教工作情况填写《信息学院研究生助教工作评定表》（简称评定表）。评定结果合格方可申请毕业答辩，评定结果不合格需重新申请担任助教工作。研究生申请毕业答辩前，提交评定表到学院备案。 | | | | | |
| **业务的数据需求** | 本人在助教系统需要的数据需求如下：   1. 志愿选择（志愿id，志愿选择时间） 2. 授课教师（授课教师id，是否为授课教师） 3. 教师（教师id，教师姓名，职称，电话，出生日期，邮箱地址，性别） 4. 研究生（研究生学号，研究生姓名，联系方式，出生日期，性别，邮箱） 5. 研究生助教评定（研究生助教评定id，助教工作自述，授课教师评价，授课教师评价结果，审核结果，审核时间） 6. 课程（课程id，课程名，课程性质，课程开始时间，课程结束时间，授课对象，选课人数，学时） | | | | | |
| **业务的非功能需求** | 安全性：   1. 登录模块需要首先在系统中登录信息，未登录成功的用户不能进入数据库进行操作。 2. 建立数据库时需要设置数据库的账户的权限，防止对于数据库的修改。 3. 在数据库建立时需要设计数据库日志信息，方便数据进行回滚。 4. 设置触发器进行数据库的数据更改的验证信息，防止数据库的数据被非法篡改。 5. 数据库进行定期备份，方便数据库出现错误时进行回滚。   完整性：   1. 实体完整性：每一个表的主键必须为独立且非空的，外键只能是null或者外键对应的表中已经存在的数据库的数据信息。 2. 参照完整性：两个表之间通过外键进行连接，防止出现一个表信息删除另一个表仍然调用信息的情况出现。 3. 用户自定义完整性：每一个表的id都设置为char类型的十位数，这样可以统一表的可用数目，方便数据库的读取。涉及钱的部分规定小数点后的位数，这样可以使表数据更加的完善。 | | | | | |
| 3、设计、开发解决方案 | | | | | | |
| **课程设计工作详述** | 本人负责的助教系统的设计如下：   1. 首先进行了业务的数据字典的抽取工作，为ER图的设计做准备，包括：志愿选择（志愿id，志愿选择时间）、授课教师（授课教师id，是否为授课教师）、教师（教师id，教师姓名，职称，电话，出生日期，邮箱地址，性别）、研究生（研究生学号，研究生姓名，联系方式，出生日期，性别，邮箱）、研究生助教评定（研究生助教评定id，助教工作自述，授课教师评价，授课教师评价结果，审核结果，审核时间）、课程（课程id，课程名，课程性质，课程开始时间，课程结束时间，授课对象，选课人数，学时）。 2. 之后开始进行了ER图的绘制工作，使用的工具为Powerdesigner，绘制的ER图如下：   ER图part1   1. 对该ER图进行了设计分析并修改之后进行了物理结构设计，根据已有的ER图之间的关系判断是否需要设计中间表，以及是否要为表添加外键。得到的关系模式如下：  |  |  |  | | --- | --- | --- | | 关系模式 | 主码 | 外码 | | 表助教选择（课程id，研究生助教评定id） | 课程id，研究生助教评定id | 课程id，研究生助教评定id | | 表志愿选择（志愿id，课程id，研究生学号，志愿选择时间） | 志愿id | 课程id，研究生学号 | | 表授课教师（授课教师id，管理员id，教师id，是否为授课教师） | 授课教师id | 管理员id，教师id | | 表授课教师教授课程（授课教师id，课程id） | 授课教师id，课程id | 授课教师id，课程id | | 表研究生（研究生学号，管理员id，教师id，研究生姓名，联系方式，出生日期，性别，邮箱） | 研究生学号 | 管理员id，教师id | | 表研究生助教评定（研究生助教评定id，管理员id，研究生学号，授课教师id，助教工作自述，授课教师评价，授课教师评价结果，审核结果，审核时间） | 研究生助教评定id | 管理员id，研究生学号，授课教师id | | 表课程（课程id，课程名，课程性质，课程开始时间，课程结束时间，授课对象，选课人数，学时） | 课程id | 无 | | 表教师（教师id，教师姓名，职称，电话，出生日期，邮箱地址，性别） | 教师id | 无 |  1. 根据关系模式设计的DDL语句汇总在总的数据库创建中，并包含数据插入语句。数据库的版本为SQL Server。 2. Dao层的测试汇总在总的Java程序中，利用Junit测试类，测试了数据的增、删、改、查方法，结果显示正确。 | | | | | |
| 4、项目管理 | | | | | | |
| **工程管理** | 1. 需求分析：输入需求，即设计一个助教管理系统，对需求进行详细分析设计，输出了助教管理系统需要的数据的数据字典。 2. 概念结构分析：输入了以及选好的数据字典，根据数据字典设计子系统的ER图。 3. 逻辑结构分析：输入ER图，根据ER图整理出助教管理系统的关系模式，当ER图出现多对多关系时，需要设置中间表来进行管理，一对一关系不需要设置中间表，但需要在两个表中互相插入对方的主键作为外键，一对多关系设计时需要在多的那个表中插入对应的一的表的主键作为外键。 4. 物理结构设计：输入关系模式，根据关系模式确定物理存储结构，利用SQL Server进行实施。 5. 实施：根据关系模式和环境信息，设计DDL语句。 | | | | | |
| **资源管理** | 在本次数据库设计中，为了实现多人协作模式，我们将整个课设从文件资源到数据库资源到持久层设计代码全部托管到了GitHub上，利用团队管理模式对GitHub资源进行了管理，具体网址设置为：  <https://github.com/Brief-drizzle/database-ks/tree/master>  为了更好的进行多人协作，我们采取了多次腾讯会议的讨论。由于没有一个合适的数据库资源服务器，故而数据库中资源的管理文件是利用微信群进行传递的，我们设置了提交时间，在规定时间内从GitHub上pull下数据库文件基本，并在规定时间内完成自己部分的DDL语句和SQL语句的设计和修改以达到课程设计的设计要求。 | | | | | |
| **风险管理** | 1. 在数据库建立时设置不同账号的账户密码以及对应的管理权限，防止非法用户的非法访问和非法操作。 2. 对数据库的主码设置非空约定，同时外码不能设置不存在的主码的值。添加外码时，要去对应主码的表中进行查找，只能插入对应的主码或空值，如果不符合要求，不能插入。 3. 对数据进行完整性约束，即主码不能为空，用户自定义约束条件，防止非法数据的插入。 4. 对数据库设置日志文件，记录数据库的操作方便出现故障时及时进行复原以及操作回滚。 5. 在用户从客户端访问数据库时要先进行用户判断，如果用户密码权限不符合不能修改数据库的内容。 6. 将数据库备份到小组每个成员的电脑上并且在github上保留最新版，防止出现数据库误操作时无法进行恢复。 | | | | | |