CS33503数据库系统实验

实验检查记录

|  |  |  |  |
| --- | --- | --- | --- |
| 实验结果的正确性(60%) |  | 表达能力(10%) |  |
| 实验过程的规范性(10%) |  | 实验报告(20%) |  |
| 加分(5%) |  | 总成绩(100%) |  |

实验报告

一、实验目的（介绍实验目的）

|  |
| --- |
|  |

二、实验环境（介绍实验使用的硬件设备、软件系统、开发工具等）

|  |
| --- |
| 硬件：x64笔记本  软件系统：windows10  开发工具 ：vscode、IDEA、SQLyog |

三、实验过程（介绍实验过程、设计方案、实现方法、实验结果等）

|  |
| --- |
| 一、数据库设计  **1.需求分析**  我本次实验是设计实现了一个简易的校园人事管理系统。  本次实验共涉及到5个实体：学生、教师、院系、班级、课程，它们的结构为：   1. 学生：包含学号、姓名、性别、所属班级的班号、入学年份； 2. 教师：包含教师编号、姓名、所属院系的编号； 3. 院系：包含院系编号、院系名； 4. 班级：包含班号、班主任老师的编号、所属院系的编号； 5. 课程：包含课程编号、课程名、授课教师的编号；   它们之间的约束关系为：   1. 学生所属班级的班号所对应的班级一定存在； 2. 教师所属院系编号对应的院系一定存在； 3. 班级班主任老师编号所对应的老师一定存在； 4. 班级所属院系编号对应的院系一定存在； 5. 课程的授课老师编号所对应的老师一定存在；   本次实验所设计到的对数据的操作为：对学生的增删改查、对教师的增和查、对院系的查、对班级的增和查、对课程的增和查。  本次实验以对学生的操作为例，来探讨通过数据库设计和索引来提高数据操作的性能。  **2.数据库设计**  用ER图来表示该数据库为：    转化为关系关系数据库模式为：    规范化：   1. 数据库的关系中所有属性都是不可再分的，满足第一范式； 2. 数据库的所有关系中，非主属性都完全函数依赖于主键，满足第二范式； 3. 数据库的所有关系中，非主属性都不传递函数依赖于候选键，满足第三范式；   物理数据库设计优化：  这里以对学生表的操作为例，来探讨如何通过数据库索引来加快数据库操作，从而提高性能，提高吞吐量。  因为查询时，主要涉及到一下查询条件：    而学号（ID）是主键，已经存在一个聚集索引了，所以这里我们为班号、入学年份这2个属性分别建立一个二级索引，然后再为姓名、班号这2个属性建立一个联合索引，从而提高查询效率。  **3.数据生成器**  通过代码的形式来向数据库中批量插入数据，具体过程如下：   1. 向数据库中插入13个提前设定好的院系； 2. 随机生成350个教师，对于每个教师来说，他的姓名是随机生成的男姓名或女性名，他所属院系的id是随机生成的； 3. 随机生成班级，对于2010-2020这21年，每年生成120个班级，所属院系id是随机的； 4. 生成60门课程，课程名是写在文件中的，授课教师是随机的； 5. 对于2010-2020这21年，每年生成5000名学生，性别是随机的，姓名是根据性别随机生成的，班号是随机的；   **4.数据库应用开发**  应用程序基于前后端分离的思想来开发，前端使用Vue相关框架来搭建，后端使用MVC模式来完成，应用效果如下：    效果图1：查询（可以加条件）    效果图2：底部分页条    效果图3：添加一条记录    效果图4：修改一条记录  **5.数据库性能优化**  这里以查询某个班级的所有学生这个功能为例，来进行数据库性能优化。  在没有进行任何优化之前，执行下面这条SQL语句：  SELECT \* FROM student WHERE class\_id=996;  它花费的时间为46ms，具体见下图：      然后我们通过为该属性添加索引的方式来优化性能，添加索引后再次执行该SQL语句的耗时为1ms（结果见下图），可以发现通过添加索引的方式将该功能的数据库部分的性能提高了40多倍。 |

四、实验结论（总结实验发现及结论）

|  |
| --- |
| 1. 数据库的设计过程是：概念数据库 -> 逻辑数据库 -> 物理数据库； 2. 规范化的数据库设计可以减少数据冗余； 3. 使用索引可以非常有效的提高数据库的查询性能； 4. 对于上层应用来说，我们应该提供优美的UI界面，从而改善用户体验； |