山东大学 软件 学院

非关系型数据库 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201600301079 | 姓名： 崔玉峰 | | 班级： 2016级软件4班 |
| 实验题目：实验六： 使用Java做选课功能(2学时) | | | |
| 实验学时：2 | | 实验日期： 2018/11/26 | |
| 实验目的  实验六： 使用Java做选课功能(2学时) 实验内容 用API方式，做学生选课功能。 重要提醒 将主要程序代码以及运行界面截图保存，用于完成实验报告。 实验要求 使用Java语言实现学生选课功能界面，根据录入的学号列出已选课程，可以更新已选课程，也可以新增选课。 | | | |
| 硬件环境：  PC  CPU : intel-i5  RAM: 8G | | | |
| 软件环境：    数据库： MongoDB  高级程序设计语言：java  IDE : Navicat for MongoDB ,IntelliJ IDEA | | | |
| 实验步骤与内容：   1. **实验概述：**   选课功能实现：   1. **根据学号找出所有已选课程。** 2. **为该学生通过CID新增课程** 3. **修改该学生任意一门选课的成绩** 4. **修改该学生的已选课程改为其他课程**   主要用到MongoDB的更新，查询两个操作，可以用之前编写的java接口进行调用。并且实现简单的交互界面，方便进行交互查看运行结果。   1. **设计思路：**   在本次实验的数据库设计上，因为考虑到MongoDB数据库不支持表连接的操作，就采用了合并关系表，引入冗余的数据，所以在进行查找时速度很快，但是更新数据时需要对多个集合进行更新。但是在总体上不用JAVA程序遍历整个集合，所以效率整体提高 。集合结构：  s**tudent 集合：**  **SID NAME SEX AGE BIRTHDAY COURSES(选课数组)**  **course 集合：**  **CID NAME FCID CREDIT STUDENTS(选课学生数组) TEACHER(任课老师)**  **teacher 集合：**  **TIDNAME SEX AGE BIRTHDAY COURSES(教课数组)**   1. **根据学号找出所有已选课程。**   通过直接在student集合中根据SID查找该学生的选课数组COURSES，就可以找到该学生所有的已选课程CID SCORE （可以根据CID再查找每个课程的具体情况）   1. **为该学生通过CID新增课程：**   更新student集合的选课数组COURSES,增加一个CID  更新course集合的STUDENTS数组，增加一个SID   1. **修改该学生任意一门选课的成绩**   更新student集合的选课数组COURSES，将该CID的课程的分数SCORE更新  更新course集合的STUDENTS数组，将该学生SID的分数SCORE更新    **④ 修改该学生的已选课程改为其他课程**    更新该学生的选课数组COURSES，将原来的课程CID改为新的课程CID。  更新旧课程的学生数组STUDENTS，将该学生的选课记录删除  更新新课程的学生数组STUDENTS，添加该学生  **数据完整性：**  对于上述的更新，插入操作都是对于多个集合的操作，但是在MongoDB中并没有事务操作，只能保证单个Update操作是原子性的。如果通过JAVA代码实现事务回滚提交有太过于复杂，所以在本实验并没有设计相应的事务机制。但是MongoDB作为一款Nosql数据库对原子性的要求不高，其优势在于json文档储存，更多应用做海量数据的缓存，查询分析。   1. **Java具体代码实现：**   **用到了以前编写的更新查询接口，具体实现思路与上述思路相同。Update语句中加入$addToSet可以向数组字段中插入数据，$pull可以从数组中删除数据组：**  **① 根据学号找出所有已选课程。**      **② 为该学生通过CID新增课程：**    **③ 修改该学生任意一门选课的成绩**     1. **修改该学生的已选课程改为其他课程**      1. **简单编写界面，输入框，按钮和输出表格，并实现简单的交互逻辑，传参调接口：**   **24BCNSK11F0D(~RWB`2I3WI**   1. **实验运行演示**   **① 输入学号进行查询已选课程：**  **测试学号200900130604:**  **A13UC20[]BE)~6EI$%$2PSB**    **测试学号201600301079：**  （这是在实验四时新加的学生所以没有选课）  PI7[`)C61)6K2C(}MCDOCFV  **② 输入课程号添加课程：**    0((S2S`8A)VL[K]U5]7VVG0  **③ 修改该学生的成绩：**  **AFH}PF9Y2$K8Y[~BWD9}5]T**  KWXI{U(PMY~_F(0D6OPFKPK  **④ 修改该学生的已选课程改为其他课程**  **G{[QOLP@HKPNK`OW_5~)VM7**  **M)%~0P~YN)ZTPKIMJ)~$[)8**  RN2G5G1_{F~XO)05S6[IKDG | | | |
| 结论分析与体会：    通过实验完成了一个拥有前端交互界面的学生选课系统，可以查询学生选课，为学生添加选课，修改选课，修改成绩，所有功能全部测试通过。有了之前对MongoDB数据库进行增删改查操作的基础，实现上述功能操作并不难，执行增删改查操作，将返回的json格式字符串，解析并显示到界面上，  主要在于设计MongoDB数据库时对于选课数据进行了冗余，所以大大减少了表连接的操作，不需要通过java for 循坏进行多次遍历，所以加快查询效率。对于更新插入操作时需要多次更新，保持数据一致性。  通过实验也体会到了关系型数据库和非关系型数据库从表设计到实现具体功能的不同，更加巩固了之前所学的知识。 | | | |