山东大学 软件 学院

非关系型数据库 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202000300125 | 姓名：贾星宇 | | 班级：2020级5班 |
| 实验题目：Mongodb的增删改查及应用 | | | |
| 实验学时：16 | | 实验日期：2022年11月10日 | |
| 实验目的：  非关系型数据库实验的Mongodb部分共有六部分实验，其中实验一二三四为熟悉其增删改查操作，实验五六为应用Mongodb实现部分场景，如学生选课及数据统计。详细目的如下：   1. 熟悉环境、建立/删除集合、插入数据(2学时)   连接MongoDB，使用命令行方式创建自己的数据库（user+学号，例如user201400300001），创建集合，输入3行数据，集合名、列名、数据值均采用英文。   1. 使用Java做简单查询(2学时)   用MongoDB API方式，做简单查询。   1. 使用Java做简单数据插入(2学时)   用MongoDB API方式，做数据插入。   1. 使用Java做数据更新(2学时)   用MongoDB API方式，做数据更新（类似SQL中的update语句功能）。   1. 使用Java做选课功能(2学时)   用MongoDB API方式，做学生选课功能。   1. 使用Java做MongoDB数据统计分析(6学时)   用MongoDB API方式，做学生数据统计分析。 | | | |
| 硬件环境：  Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz | | | |
| 软件环境：  Intellij IDEA+MongoDB | | | |
| 实验步骤与内容：   1. 实验一：熟悉环境，建立/删除集合、插入数据    * + 1. 连接MongoDB，下载压缩包，解压后建立相应文件夹，如db等。随后输入命令：D:\MC\_00\MongoDB\bin\mongod.exe --dbpath D:\MC\_00\MongoDB\data\db 以启动mongodb      * + - 1. 创建学生集合、教师集合、课程集合：为方便操作，后续语句运行及数据库内容的查看使用工具Studio 3T：     创建数据库的语法如下：  Use DB\_NAME    因此，使用use student等sql创建相应数据库。   * + - 1. 插入数据：   插入数据的语法为：  db.COLLECTION\_NAME.insert(document)  或  db.COLLECTION\_NAME.save(document)  因此，将要求表格数据转换为JSON格式，并依次执行：  db.student.insert(  [  {"sid":"200800020101","name":"Wang Xin","sex":"女","age":"21","birthday":"1994-2-2","dname":"CS","class\_":"2010"},  {"sid":"200800020102","name":"Li Hua","sex":"女","age":"20","birthday":"1995-3-3","dname":"SC","class\_":"2009"},  {"sid":"200800020103","name":"Zhao Yan","sex":"男","age":"19","birthday":"1996-4-4","dname":"SC","class\_":"2009"}  ]  )  db.course.insert(  [  {"cid":"300001","name":"DS","fcid":"","credit":"2"},  {"cid":"300002","name":"DB","fcid":"300001","credit":"2.5"},  {"cid":"300003","name":"OS","fcid":"300001","credit":"4"}  ]  )  db.teacher.insert(  [  {"tid":"100101","name":"Zhang","sex":"F","age":"44","dname":"CS"},  {"tid":"100102","name":"Li","sex":"M","age":"45","dname":"SC"},  {"tid":"100103","name":"Ma","sex":"M","age":"46","dname":"CS"}  ]  )  student\_course、teacher\_course同理。   1. 实验二：使用JAVA做简单查询   从此实验开始，本实验的查询及显示皆通过前后端交互的方式完成，其中前端使用vue框架，并利用vuetify插件进行装饰；后端使用springboot框架，并利用mybatisPlus与mongoDB连接。  利用springboot运行mongodb使用了MongoTemplate，查询条件利用Query和Criteria类进行。  绘制前端界面（整个实验）：    其中，左侧红色区域为可以执行的操作，点击按钮以执行；右侧为显示页面，显示执行结果或进行进一步的相关操作。  1.开始实验，首先选择题目“找出所有学生关系的所有学生”，可以发现需要从student表中检索到所有数据并返回给前端页面。后端代码如下：    利用findAll方法便可以列表的形式返回student表中的所有数据。前端 接收到的参数如下：    随后在前端利用v-chip插件和v-for循环显示相关内容，做到点击按钮时，将学生信息显示在右侧页面：    2.利用类似的方法实现查询所有老师、所有课程：    （图：查询所有老师）    （图：查询所有课程）  3.随后，实现查询年龄小于20岁的学生，此时需要用到Query类的限制条件，具体语法如下：    其中关键语法为：  Criteria.*where*("AGE").lt(20)  点击前端相应按钮，将调用此接口方法，并将list类型的数据返回至前端显示：    可以发现，所查询学生年龄皆小于20岁。  4.同理，实现年龄小于20岁且学院为软件学院的学生。后端代码：    前端显示：    可以发现满足要求。  5.随后，实现了查询所有学生姓名和性别的功能：  后端利用include()方法实现       1. 实验三：使用Java做简单数据插入。   此次实验核心内容为插入，本人继续使用前后端交互的方式完成实验。  1.学生数据的插入：  前端使用v-text-field插件绘制输入框，利用v-model动态绑定数据。点击左侧按钮“向学生表插入数据”后，右侧便弹出输入框与输入确定框：    用户输入相应数据后，点击确认按钮，数据便会保存至后端。其中后端代码如下：    检查结果是否插入：    值得注意的是，系统支持表格插入。点击蓝色“上传”按钮后，系统要求选中相应表格，并将表格数据转为JSON格式向后端传输，后端接受到数据后保存到相应document中：    （图：点击上传按钮选择文件）    （图：解析好的JSON格式）  2.教师表数据插入：  同上，前端交互界面输入数据，后端存储：    （图：前端上传数据）    （图：后端处理语句）    （图：存储结果）  3.课程表内容插入：    （图：前端上传数据）    （图：后端处理语句）    （图：存储结果）   1. 实验四 使用Java做数据更新   本人将数据插入及数据更新整合到同一界面完成，即在插入学生、课程或老师时，如果其id没有出现在数据库中，则执行插入操作；如果其id出现在了数据库中，则执行更新操作进行更新，更新内容为id外其他内容。上述操作在后端自动完成，对用户透明。  1.学生数据的更新：  前端使用v-text-field输入数据，随后将学生数据打包好以json的格式传送至后端。后端代码如下：    首先利用mongoTemplate的find方法搜索学号为students学号的元组，如果搜索到的内容为空，则执行insert方法做实验三的插入操作，如果不为空则开始更新：首先创建以sid为搜索条件的Query，随后创建更新sid之外数据的update，并利用updateFirst方法更新。具体表现如下，如将学号为202000300125的学生的生日更新为2002.03.05：    更新前：    更新后：    同理，更新teacher和course的后端java语言如下：  Teacher：    Course：     1. 实验五 使用Java做选课功能   1.首先展示前端页面及操作逻辑。点击按钮“学生选课”，右侧弹出界面如下：    输入学号，点击“查询已选课程”按钮，用户可以浏览到自己已选择的课程：    学生选课功能：点击下方灰色课程，下方红色行会提示选中某节课，点击选中确认后，表示此学生已选中此门课程，上方红色条目会显示新选中的课程。如选择“高级程序语言”课程：    删除（修改）选中课程:点击红色“X”号可删除相应已选中课程，并进行重选，如点击“老庄导读”课程右侧红色按钮：    可以看到此课程已删除。  2.选课后端逻辑：  （1）首先，为了学生能够选课，需要查询到学生可以选择的所有课程，因此需要在点击“学生选课”按钮后前端需要隐式调用后端接口得到可选的所有课程：    随后，学生输入学号后，后端在student\_course表中根据学号找到学生选择的所有cid：    由于先前前端已经存储到了所有的课程信息，因此将cid与课程name进行连接的操作我们放到前端进行：    如图，利用两层for循环，将学生已选中的课程的cid和name等信息存储到course数组中。  （2）学生选课。前端学生选中相应课程后，利用v-for循环和动态绑定的特性将所选中课程的课程号和学生学号发送至后端。后端找到对应老师后加上老师工号，并将数据存储至student\_course表中：    （3）删除学生所选课程：前端点击X号后，通过动态绑定的方式获得相应学生学号和课程号，随后发送至后端。后端利用remove方法将所选中数据删除：     1. 实验六 使用Java做MongoDB数据统计分析   1.列出student\_course集合中出现过的所有课程名称。  重点关注的问题为去重问题。前端点击相应按钮后，右侧便会以课程号-课程名称的方式列出集合中出现的所有课程：    当点击前端按钮时，执行两个方法，首先后端执行如下函数：    通过此方法，前端将得到存在的所有课程信息，包含课程号、课程名称等。随后前端将所有课程信息存储到相应数组中。  随后调用如下方法：    函数通过findDistinct方法找到student\_course集合中出现过的所有课程编号，且将重复编号去掉并将编号返回给前端。  在前端进行连接操作，利用for循环依次找到每个课程编号对应的课程名称，并依次存储到course数组中：    2.找出平均成绩排名前十的学生。  效果如下：前端点击按钮，可以计算平均成绩并列出平均成绩排名前十的学生信息：    前端向后端发起请求，后端执行如下操作：    首先通过Aggregation类执行聚合函数，主要做如下事情：  （1）通过SID进行分组  （2）对每一组学生成绩求平均值，并命名为avgscore  （3）按照avgscore对分组结果降序排列  （4）通过getMappedResults方法得到相应结果。  （5）针对每个结果的SID，在student类中进行查询，并将查询到的结果加上平均成绩 存储到result列表中  （6）返回  返回结果大致结构：    随后，利用前端v-for循环将结果显示在页面上，完成实验。  至此，实验一到实验六完成。 | | | |
| 结论分析与体会：  通过对这六个实验的实操完成，我对非关系型数据库MongoDB在数据库创建、增删改查、数据库应用以及与springboot框架结合等多个方面有了非常清晰和明确的认识，同时我也发现了自己固有知识的不足，自己的思维时长限制在关系型数据库的认识中无法自拔，因此初始使用MongoDB时有些许的不适应，但在逐渐使用的过程中也认识到了非关系型数据库的优势和亮点。  在日后的学习中，我的思维将更加开阔明朗，将愈发熟练地利用多方面知识解决现实中的问题，变为一个更有用的人！ | | | |