数据库原理 第三次实验报告

09019106 牟倪

# 一、实验目的

* 了解ER图的绘制方法。
* 使用关系模型对现实世界进行建模，加深对关系模型的理解。
* 巩固Access基本操作。
* 熟练掌握Access中的SQL查询。

# 二、实验内容

## 2.1 应用背景 & 关系模型的设计

隔壁的数字图像处理课堂，经常会进行小测验，测验题目为不定项选择题。通过抽象出“小测验”这一场景涉及的实体（entity）与关系（relationship）、建立关系模型，我们可以对测验题目、作答情况等相关信息进行管理。

实体：

* 老师：包含工号（整数）、姓名（字符串）、学院（字符串）、邮箱（字符串）属性，其中工号可以唯一确定一位老师。
* 学生：包含学号（整数）、姓名（字符串）、班级（字符串）属性，其中学号可以唯一确定一位同学。
* 题目：包含题目编号（整数）、题干（字符串）、4个选项（ABCD选项，字符串）、正确选项（ABCD组成的字符串）属性，其中题目编号可以唯一确定一道题目。

关系：

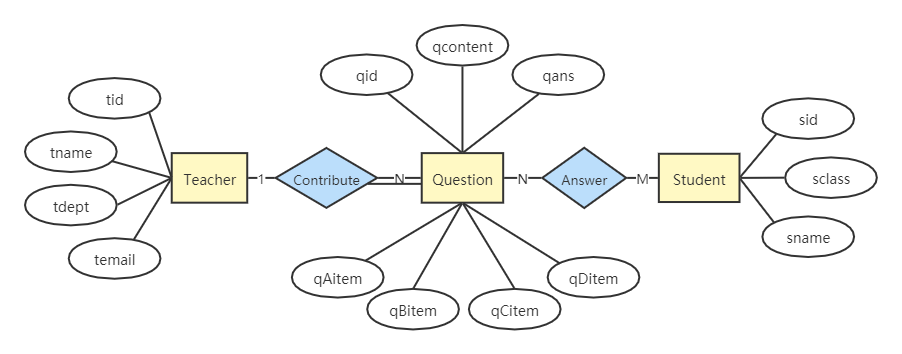
* 一个题目由一位老师贡献，每个老师可以贡献多个题目。
* 学生可以作答多道题目，一道题目可以被多个学生作答，学生不能重复作答同一道题目。

我们考虑建立以下5张表：

* 老师信息表：Teachers(tid, tname, tdept, temail)，属性分别为工号、姓名、学院、邮箱，tid为主键。
* 学生信息表：Students(sid, sname, sclass)，属性分别为学号、姓名、班级，sid为主键。
* 题目信息表（老师贡献题目的关联表）：Questions(qid, tid, qcontent, qAitem, qBitem, qCitem, qDitem, qans)，属性分别为题目编号、贡献老师的工号、题干、A选项内容、B选项内容、C选项内容、D选项内容、题目答案，qid为主键。
* 学生作答题目的关联表：Answers(sid, qid, sans, qans, correct)，属性分别为学生学号、题目编号、学生答案、题目答案、是否正确（boolean值），sid、qid共同作为主键。

## 2.2 ER图的设计

可以看出，“老师贡献题目”是1:N的关系，“学生作答题目”是N:M的关系；每一个题目都必须被老师贡献，否则就不是合法的题目。由此，我们绘制ER图：



## 2.3 功能的设计与SQL实现

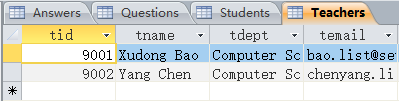
|  |  |  |
| --- | --- | --- |
| 函数命名 | 功能描述 | SQL语句（示例数据） |
| insertTeacher | 新增老师信息 | INSERT INTO  Teachers(tid, tname, tdept, temail)  VALUES(9001, 'Xudong Bao', 'Computer Science', 'bao.list@seu.edu.cn'); |
| insertStudent | 新增学生信息 | INSERT INTO  Students(sid, sname, sclass)  VALUES(213190001, 'Tom', '090191'); |
| insertQuestion | 新增题目信息 | INSERT INTO  Questions(qid, tid, qcontent, qAitem, qBitem, qCitem, qDitem, qans)  SELECT  MAX(qid)+1,  9001,  'gamma灰度映射中，可以实现低灰度段拉伸的gamma取值有',  '0.1',  '0.2',  '1',  '3',  'AB'  FROM Questions; |
| queryTeacherId | 根据工号查询老师 | SELECT \* FROM Teachers WHERE tid=9001; |
| queryTeacherName | 根据姓名查询老师 | SELECT \* FROM Teachers WHERE tname='Xudong Bao'; |
| queryStudentId | 根据学号查询学生 | SELECT \* FROM Students WHERE sid=213190001; |
| queryStudentName | 根据姓名查询学生 | SELECT \* FROM Students WHERE sname='Tom'; |
| queryQuestionId | 根据题目编号查询题目 | SELECT \* FROM Questions WHERE qid=2; |
| queryQuestionTeacher | 根据贡献老师查询题目 | SELECT \* FROM Questions WHERE tid=9001; |
| queryQuestionContent | 根据题目内容模糊查询题目 | SELECT \* FROM Questions  WHERE qcontent LIKE '\*二值图像\*'; |
| insertAnswer | 新增学生作答记录 | INSERT INTO Answers(sid, qid, sans, qans, correct)  VALUES(213190001, 1, 'BC', 'BC', true);  由于技术限制，需要将题目标准答案与是否正确作为参数输入。 |
| queryAnswerQ | 根据题目编号查询作答记录 | SELECT \* FROM Answers WHERE qid=1; |
| queryAnswerS | 根据学号查询作答记录 | SELECT \* FROM Answers WHERE sid=213190001; |
| queryAnswerQS | 根据学号与题目编号查询作答记录 | SELECT \* FROM Answers  WHERE qid=1 AND sid=213190001; |
| getAvgQCorrectness | 得到某题目的正答率 | SELECT -SUM(correct)/COUNT(\*) AS avg  FROM Answers  WHERE qid=1; |
| getAvgStuCorrectness | 得到某学生的平均正答率 | SELECT -SUM(correct)/COUNT(\*) AS avg  FROM Answers  WHERE sid=213190001; |
| getStuAnsAllQuestion | 得到做完所有题目的学生的学号列表 | SELECT DISTINCT Answers.sid FROM Answers  WHERE NOT EXISTS(  SELECT Questions.qid FROM Questions  WHERE Questions.qid NOT IN(  SELECT A2.qid FROM Answers AS A2  WHERE A2.sid=Answers.sid  )); |

## 2.4 实验结果

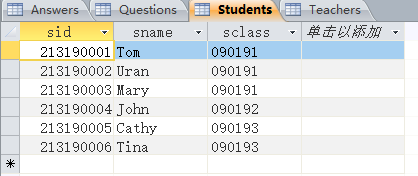
### 2.4.1插入数据

以下为执行4个insert查询数次后，4张表的结果：

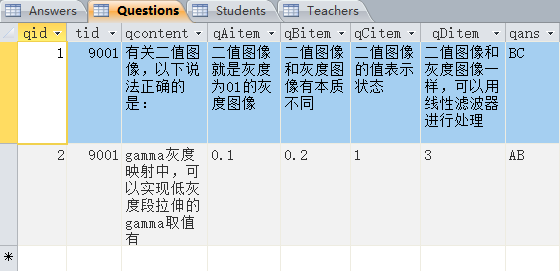
Teachers：



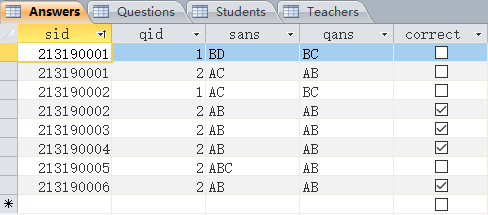
Students：



Questions：

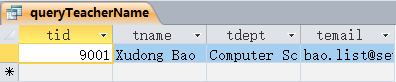


Answers：

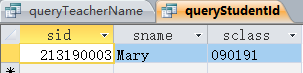


### 2.4.2查询数据

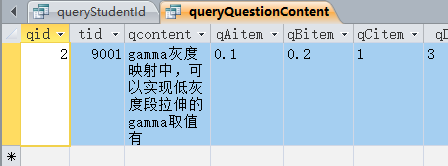
以“Xudong Bao”为参数按姓名对老师进行查询：



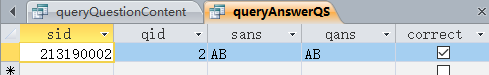
以213190003为参数按学号对学生进行查询：



以“灰度”参数对题目进行模糊查询：

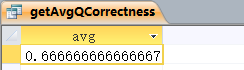


以213190002的学号、2的题目编号，对作答记录进行查询：

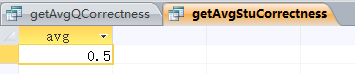


### 2.4.3 集合函数查询

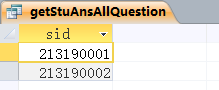
查询编号为2的题目的正答率：



查询学号为213190002的学生的平均正答率：



查询做完所有题目的学生：



# 三、实验总结

通过本次实验，我学习了ER图的绘制方法，并对身边的场景进行关系模型建模，对关系模型的理解更加深刻。在代码编写的过程中，我深深感受到，虽然SQL编程的基本语法和思想是统一的，但语言细节因平台而异；理清逻辑、写出伪代码并不困难，但大多数时间都花费在查阅Access文档上（尤其是编写INSERT语句嵌套SELECT语句时）。