**数据库实验**

**第一次实验——SQL语言中的单表查询与连接查询**

曹广杰

数据科学与计算机学院

2017/9/30

任课教师：刘玉葆。

关键词：数据库；SQL语言；连接查询；

Contents

[Task1 1](#_Toc494614484)

[实验结果： 2](#_Toc494614485)

[Task2 2](#_Toc494614486)

[实验结果 3](#_Toc494614487)

[Task3 3](#_Toc494614488)

[实验结果： 4](#_Toc494614489)

[Task4 4](#_Toc494614490)

[实验结果： 4](#_Toc494614491)

[Task5 4](#_Toc494614492)

[实验结果： 5](#_Toc494614493)

[Task6 5](#_Toc494614494)

[实验结果： 6](#_Toc494614495)

[Task7 6](#_Toc494614496)

[实验结果： 6](#_Toc494614497)

[Task8 7](#_Toc494614498)

[实验结果： 7](#_Toc494614499)

[Task9 7](#_Toc494614500)

[实验结果： 8](#_Toc494614501)

[Task10 8](#_Toc494614502)

[实验结果： 8](#_Toc494614503)

本次实验基于school数据，联系SQL查询语言进行实验。该数据文件由4种关系构成，分别为courses、students、choices 和 teachers；

## Task1

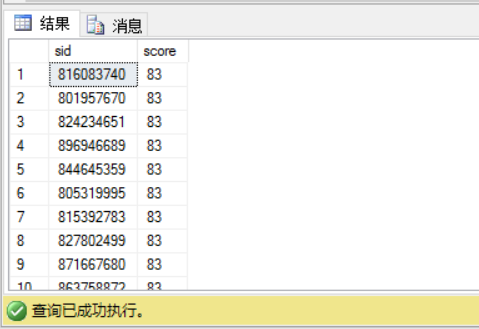
要求：查询选修C++课程的成绩比姓名为 ZNKOO的学生高的所有学生的编号和姓名。

|  |
| --- |
| **-- 查找c++科目的成绩**  **select students.sid, score**  **from CHOICES, STUDENTS, COURSES**  **where STUDENTS.sid = CHOICES.sid**  **and COURSES.cid = CHOICES.cid**  **and cname = 'c++'**  **-- 分数比znkoo高的限定条件**  **and score >**  **(select score -- znkoo的分数**  **from CHOICES, STUDENTS, COURSES**  **where STUDENTS.sid = CHOICES.sid**  **and COURSES.cid = CHOICES.cid**  **and sname = 'znkoo'**  **and cname = 'c++')**  **order by score** |

句法：

1. 结构：此处使用子查询语言，在表达znkoo的分数时候，由于根据学生姓名不能直接获得其分数，故而需要查询。在这种条件下，使用子查询语言；
2. 对于子查询语句，可以使用在where中附加的条件语句限定，但是在添加参与查询的对象时，有时候系统需要查询几个对象交集的笛卡尔集，这使得计算复杂度直线上升。而使用子查询语言的时候，则分割为两个查询任务，此时，要注意子查询语言的返回值应该有且只有明确的一个值。

### 实验结果：



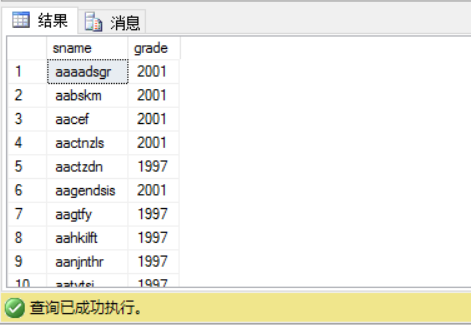
## Task2

要求：找出和学生883794999或学生850955252的年级一样的学生的姓名。

|  |
| --- |
| **-- 其中一个所在的年级**  **select sname, grade**  **from STUDENTS**  **where grade = (**  **select grade**  **from STUDENTS**  **where sid = '883794999')**  **union**  **-- 另一个所在的年级**  **select sname, grade**  **from STUDENTS**  **where grade = (**  **select grade**  **from STUDENTS**  **where sid = '850955252')** |

查询两个不同的条件，不应该使用‘或’语言——带有比较运算符的子查询，该子查询必须返回单值，否则引起编译错误。所以，使用或语言的时候，返回值会不止一个，查询系统无从辨认。

### 实验结果



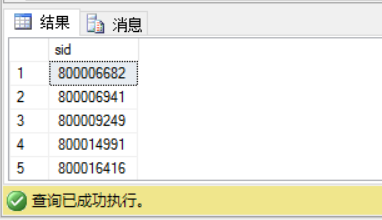
查询结果显示，两个年份分别为2001 & 1997。

## Task3

要求：查询没有选修Java的学生名称。

|  |
| --- |
| **-- 查询所有的学生**  **select sid**  **from STUDENTS**  **-- 排除选择Java的学生**  **where sid not in(**  **select sid -- 选择Java**  **from CHOICES, COURSES**  **where CHOICES.cid = COURSES.cid**  **and COURSES.cname = 'java')**  **使用not in语句，表明“不在……范围中”** |

### 实验结果：

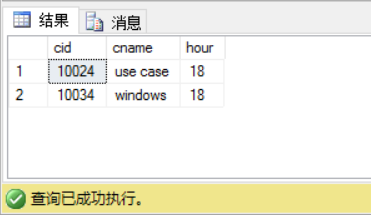


## Task4

要求：找出课时最少的课程的详细信息。

|  |
| --- |
| **select \***  **from COURSES**  **where COURSES.hour <=**  **-- 子查询， 小于每一个**  **all(select hour**  **from COURSES**  **where hour is not null)** |

### 实验结果：



可以看到，最少课时为18。

## Task5

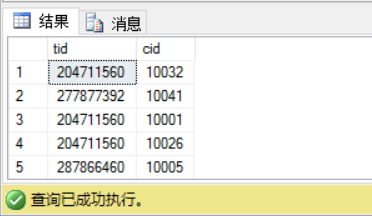
要求：查询工资最高的教师的编号和开设的课程号。

|  |
| --- |
| **-- 教师ID 与 课程ID一一对应**  **select TEACHERS.tid, cid**  **from TEACHERS, CHOICES**  **where TEACHERS.tid = CHOICES.tid**  **-- 限定条件：工资最高**  **and salary >= all(**  **select salary**  **from TEACHERS**  **where salary is not null)** |

句法：

条件 All（集合），表示集合内的所有数据都满足条件。

### 实验结果：

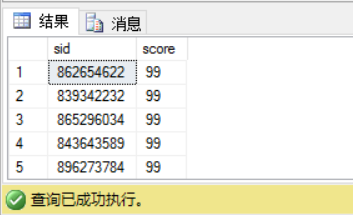


## Task6

要求：找出选修课程ERP成绩最高的学生编号。

|  |
| --- |
| **-- 成绩信息与课程信息锁定**  **select sid, score**  **from CHOICES, COURSES**  **where CHOICES.cid = COURSES.cid**  **and COURSES.cname = 'erp'**  **-- 大于所有科目最高成绩，**  **-- 则必然是erp的最高成绩**  **and score >= all**  **(select score**  **from CHOICES**  **where score is not null)** |

### 实验结果：

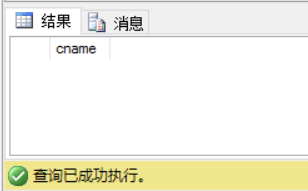


## Task7

要求：查询没有学生选修的课程名称。

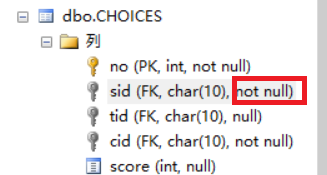
|  |
| --- |
| **select cname**  **from COURSES**  **where cid not in**  **(select cid**  **from CHOICES)**  **查询选课列表中不存在的课程。** |

### 实验结果：



查询学生的考核信息：

现在显示结果是，所有的课程都出现在选课列表中。而选课列表中，每一门课都有学生选修。



这就说明，不存在没有学生选修的课程。

即：

1. 所有课程都在选课列表中；
2. 选课列表的课都有学生选修；

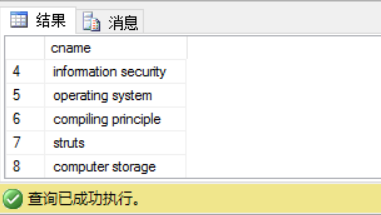
则，所有课程都有学生选修——即，不存在没有学生选修的课程。

## Task8

要求：查询讲授课程UML的教师所讲授的所有课程名称。

|  |
| --- |
| **-- 课程ID与老师ID对应**  **select cname**  **from COURSES, CHOICES**  **where CHOICES.cid = COURSES.cid**  **-- 教授UML课程的老师ID**  **and CHOICES.tid =**  **ANY(select tid**  **from CHOICES**  **-- UML课程ID**  **where cid =**  **(select cid**  **from COURSES**  **where cname = 'uml'))** |

### 实验结果：



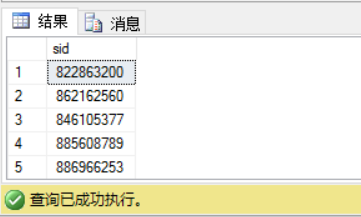
关联嵌套的信息查询，需要的信息都不易获得，在此前提下使用子查询语句逐一获得需要的信息即可，注意使用any/all/not in等关系谓词。

## Task9

要求：使用集合交运算，查询既选修了database又选修了UML课程的学生编号。

|  |
| --- |
| **-- database的查询**  **select sid**  **from CHOICES, COURSES**  **where cname = 'database'**  **and CHOICES.cid = COURSES.cid**  **intersect -- 巴啦啦能量！连接！**  **-- UML的查询**  **select sid**  **from CHOICES, COURSES**  **where cname = 'uml'**  **and CHOICES.cid = COURSES.cid** |

### 实验结果：



实际上就是分别查询，再使用一个连接词。

## Task10

要求：使用集合减运算，查询选修了database却没有选修UML课程的学生编号。

|  |
| --- |
| **-- database的查询**  **select sid**  **from CHOICES, COURSES**  **where cname = 'database'**  **and CHOICES.cid = COURSES.cid**  **except -- 巴啦啦能量！减法！**  **-- UML的查询**  **select sid**  **from CHOICES, COURSES**  **where cname = 'uml'**  **and CHOICES.cid = COURSES.cid** |

表明，选择了database，但是没有选择UML的学生而已。对此，使用except语句，进行子集的分割清除。

### 实验结果：

