[【Advanced SQL-1 】](http://blog.csdn.net/cryssdut/article/details/49302205)

这次还是之前的**[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "_blank)**，以下五道题对应教材第三章结尾部分

Using the university schema that you have write the following queries. In some cases you might need to insert extra data to show the effect of a particular feature.

//有些时候你需要修改数据库

第一题：

Insert each student as an instructor of department ‘拳脚学院’, with salary=40000

插入操作，没什么好说的，注意插之前判断一下教师表里是否已经存在这个人了

**[sql]** [view plain](http://blog.csdn.net/cryssdut/article/details/49302205) [copy](http://blog.csdn.net/cryssdut/article/details/49302205)

[print?](http://blog.csdn.net/cryssdut/article/details/49302205)

1. **insert** **into** instructor
2. **select** S.ID, S.**name**, '拳脚学院', 40000
3. **from** student S
4. **where** S.ID not in ( **select** ID **from** instructor );

第二题：

Now delete all the newly added "instructors" above (note: already existing instructors who happened to have salary=40000 should not get deleted)

删掉第一个问插入的数据

**[sql]** [view plain](http://blog.csdn.net/cryssdut/article/details/49302205) [copy](http://blog.csdn.net/cryssdut/article/details/49302205)

[print?](http://blog.csdn.net/cryssdut/article/details/49302205)

1. **delete** **from** instructor
2. **where** ID in ( **select** ID **from** student ) and
3. dept\_name = '拳脚学院' and
4. salary = 40000;

第三题：

Update the salary of each instructor to 10000 times the number of course sections they have taught.

将每个讲师的工资更新为：他所教section数 \* 10000

**[sql]** [view plain](http://blog.csdn.net/cryssdut/article/details/49302205) [copy](http://blog.csdn.net/cryssdut/article/details/49302205)

[print?](http://blog.csdn.net/cryssdut/article/details/49302205)

1. **update** instructor
2. **set** salary = 10000 \* (
3. **select** COUNT(\*)
4. **from** teaches
5. **where** teaches.ID = instructor.ID
6. )

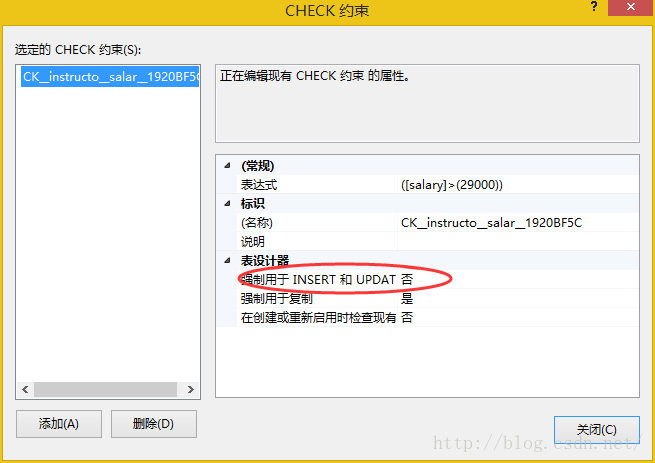
直接执行代码，会发生错误：“UPDATE语句与\*\*\*约束冲突”，原因是讲师表里对salary属性设置了CHECK约束，必须是numeric(8, 2)，10000可能是默认int型并不符合规范，查阅微软MSDN提供的官方说明：

<https://msdn.microsoft.com/zh-cn/library/aa292216(VS.71).aspx>

我们可以获得解决这一问题的方法，如下：

右键数据库设计中发生CHECK冲突的列，选择CHECK约束

将下图所示项设为“否”



再次执行代码，操作成功！

第四题：

The university rules allow an F grade to be overridden by any pass grade (for example, A). Now, lists students who have fail grades that have not been overridden. For each student as such, information displayed (in one row) should involve:

·Identifier of student

·Name of student

·Count of F grades that have not been overridden.

找出那些挂科了，并且补考也没过或者还没参加补考的人，以及他们各自挂了几科。

统计出通过课程表，那些得了F并且不在通过课程表里的，这些就是补考仍然没过或者还没补考的。最后再COUNT统计数目即可。

**[sql]** [view plain](http://blog.csdn.net/cryssdut/article/details/49302205) [copy](http://blog.csdn.net/cryssdut/article/details/49302205)

[print?](http://blog.csdn.net/cryssdut/article/details/49302205)

1. **with** pass(ID, course\_id) **as**
2. (
3. **select** **distinct** S1.ID, T1.course\_id
4. **from** student S1, takes T1
5. **where** S1.ID = T1.ID and
6. T1.grade != 'F'
7. ),
8. not\_pass(ID, **name**, course\_id) **as**
9. (
10. **select** S.ID, S.**name**, T.course\_id
11. **from** student S, takes T
12. **where** S.ID = T.ID and
13. T.grade = 'F' and
14. S.ID not in (
15. **select** pass.ID **from** pass
16. **where** T.course\_id = pass.course\_id
17. )
18. )
19. **select** ID, **name**, COUNT(\*) **as** '未通过科目数'
20. **from** not\_pass
21. **group** **by** ID, **name**;

第五题：

In one result, list the instructors who have never taught any courses and the students who have never registered for any courses. For each person, information displayed (in one row) should involve:

·Id of the person

·Name of the person

·Role of the person. The value of role should be ‘student’ or ‘instructor’.

找出那些一门课都没有选的学生，和一门课都不教的讲师，显示在同一个表里。

各自找出来后，使用集合并运算“union”即可。

**[sql]** [view plain](http://blog.csdn.net/cryssdut/article/details/49302205) [copy](http://blog.csdn.net/cryssdut/article/details/49302205)

[print?](http://blog.csdn.net/cryssdut/article/details/49302205)

1. **select** S.ID, S.**name**, 'student' **as** 'Role'
2. **from** student S
3. **where** S.ID not in ( **select** ID **from** takes)
4. **union**
5. **select** I.ID, I.**name**, 'instructor' **as** 'Role'
6. **from** instructor I
7. **where** I.ID not in ( **select** ID **from** teaches)