

work-sql

余旺

April 2024

需要说明一般而言，对于数据集都会在每道题中给出，若未给出一般情况下代表之前已经提交过了

- 1 创建一个能向学生表 Student 中插入一条记录的存储过程 Insert_Student, 该过程需要 5 个参数, 分别用来传递学号、姓名、性别、出生日期、班级号。写出调用存储过程 Insert_Student 的 Sql 语句, 向数据表 Student 中插入一个新同学, 并提供相应的实参值。

```
-- (1) 创建一个能向学生表Student中插入一条记录的存储过程Insert_Student。该过程需要5个参数，分别用来传递学号、姓名、性别、出生日期、班级号。
-- 写出调用存储过程Insert_Student的Sql语句，向数据表Student中插入一个新同学，并提供相应的实参值。
DELIMITER //
CREATE PROCEDURE Insert_Student (
    IN p_Sno VARCHAR(8),
    IN p_Sname VARCHAR(8),
    IN p_Ssex NVARCHAR(2),
    IN p_Sbirthday DATE,
    IN p_Classno VARCHAR(4)
)
BEGIN
    INSERT INTO student (Sno, Sname, Ssex, Sbirthday, Classno)
    VALUES (p_Sno, p_Sname, p_Ssex, p_Sbirthday, p_Classno);
    SELECT 'Student inserted successfully.';
END //
DELIMITER ;
CALL Insert_Student( p_Sno: '22309180', p_Sname: '张三', p_Ssex: '男', p_Sbirthday: '2000-05-15', p_Classno: 'RJ2201');
```

Student inserted successfully. *

1 Student inserted successfully.

	Sno	Sname	Ssex	Sbirthday	Classno	Totalcredit	Createdby	Age	Addr
1	22301056	余旺	男	2004-04-26	RJ2202	0	22301056	19	<null>
2	22309010	李一在	男	2005-10-01	RJ2201	0	22301056	18	<null>
3	22309012	葛杨	男	2006-08-08	RJ2201	0	22301056	18	<null>
4	22309015	刘三晶	女	2005-05-22	RJ2201	0	22301056	18	<null>
5	22309020	杨四敏	女	2007-01-08	RJ2201	0	22301056	18	<null>
6	22309030	胡五斌	男	2006-10-08	RJ2201	0	22301056	18	<null>
7	22309048	赵陈毅泽	男	2005-06-06	RJ2202	0	22301056	18	<null>
8	22309050	王威	男	2005-06-10	RJ2202	0	22301056	18	<null>
9	22309067	赵六玮	女	2005-08-21	RJ2203	0	22301056	18	<null>
10	22309075	王娜	女	2006-09-23	RJ2203	0	22301056	18	<null>
11	22309088	秦琪键	男	2005-03-01	RJ2203	0	22301056	18	<null>
12	22309100	田仪	女	2006-02-26	RJ2204	0	22301056	18	<null>
13	22309148	赵八砚	男	2005-04-25	RJ2205	0	22301056	18	<null>
14	22309150	杨青	女	2005-11-15	RJ2205	0	22301056	18	<null>
15	22309160	杨久玲	女	2005-12-12	RJ2206	0	22301056	18	<null>
16	22309180	张三	男	2000-05-15	RJ2201	0	22301056	18	<null>

- 2 创建一个向课程表 Course 中插入一门新课程的存储过程 Insert_Course，该存储过程需要三个参数，分别用来传递课程号、课程名、学分，但允许参数“学分”的默认值为 4，即当调用存储过程 I 时，未给第三个参数“学分”提供实参值时，存储过程将按默认值 4 进行运算。

调用存储过程 Insert_Course，向课程表 Course 中插入一门新课程。分两种情况（给出第三个参数和未给出第三个参数）写出相应的 Sql 命令，并比较结果。

注意：对于 MySQL，存储过程不支持函数默认参数，不支持类似函数重载的语法，可通过两个不同名存储过程来实现本题。

```
-- 注意：对于 MySQL，存储过程不支持函数默认参数，不支持类似函数重载的语法，可通过两个不同名存储过程来实现本题。
DELIMITER //
CREATE PROCEDURE Insert_Course (
    IN p_Cno VARCHAR(6),
    IN p_Cname VARCHAR(30),
    IN p_Ccredit SMALLINT
)
BEGIN
    IF p_Ccredit IS NULL THEN
        SET p_Ccredit = 4;
    END IF;

    INSERT INTO course (Cno, Cname, Ccredit)
    VALUES (p_Cno, p_Cname, p_Ccredit);
    SELECT 'Course inserted successfully.';
END //
DELIMITER ;

-- 给出第三个参数
CALL Insert_Course( p_Cno: '810020', p_Cname: '数据挖掘', p_Ccredit: 3);

-- 未给出第三个参数，使用默认值
CALL Insert_Course( p_Cno: '810021', p_Cname: '机器学习', p_Ccredit: NULL);

Course inserted successfully.
1 Course inserted successfully.
```

WHERE ORDER BY

	Cno	Cname	Ccredit
1	800001	程序设计基础	4
2	800002	面向对象程序设计	4
3	800003	数据结构	4
4	810011	数据库系统	4
5	810013	计算机网络	3
6	810015	计算机组成原理	4
7	810020	数据挖掘	3
8	810021	机器学习	4

- 3 (3) 创建一个名称为 Query_Student 的存储过程，该存储过程的功能是从数据表 Student 中根据学号查询某一同学的姓名、性别、出生日期、班级号。调用存储过程 Query_Student，查询学号为“22309150”的姓名、性别、出生日期、班级号，并在过程里输出相关结果信息。编写此功能的测试 Sql 命令。

```
876
877 -- (3) 创建一个名称为Query_Student的存储过程，该存储过程的功能是从数据表Student中根据学号查询某一同学的姓名、性别、出生日期、班级号。
878 -- 调用存储过程Query_Student，查询学号为“22309150”的姓名、性别、出生日期、班级号，并在过程里输出相关结果信息。编写此功能的测试Sql命令。
879 DELIMITER //
880
881 CREATE PROCEDURE Query_Student (
882     IN p_Sno VARCHAR(8)
883 )
884 BEGIN
885     SELECT Sname, Ssex, Sbirthday, Classno
886     FROM student
887     WHERE Sno = p_Sno;
888 END //
889
890 DELIMITER ;
891
892 CALL Query_Student( p_Sno: '22309150');
```

Sno	Sname	Ssex	Sbirthday	Classno
1	杨青	女	2005-11-15	RJ2205

Sno	Sname	Ssex	Sbirthday	Classno	Totalcredit	Createdby	Age	Addr	
1	22301056	余旺	男	2004-04-26	RJ2202	0	22301056	19	<null>
2	22309010	李一在	男	2005-10-01	RJ2201	0	22301056	18	<null>
3	22309012	葛畅	男	2006-08-08	RJ2201	0	22301056	18	<null>
4	22309015	刘三晶	女	2005-05-22	RJ2201	0	22301056	18	<null>
5	22309020	杨四敏	女	2007-01-08	RJ2201	0	22301056	18	<null>
6	22309030	胡五斌	男	2006-10-08	RJ2201	0	22301056	18	<null>
7	22309048	赵陈毅泽	男	2005-06-06	RJ2202	0	22301056	18	<null>
8	22309050	王威	男	2005-06-10	RJ2202	0	22301056	18	<null>
9	22309067	赵六玮	女	2005-08-21	RJ2203	0	22301056	18	<null>
10	22309075	王娜	女	2006-09-23	RJ2203	0	22301056	18	<null>
11	22309088	蔡琪健	男	2005-03-01	RJ2203	0	22301056	18	<null>
12	22309100	田仪	女	2006-02-26	RJ2204	0	22301056	18	<null>
13	22309148	赵八枫	男	2005-04-25	RJ2205	0	22301056	18	<null>
14	22309150	杨青	女	2005-11-15	RJ2205	0	22301056	18	<null>
15	22309160	杨久玲	女	2005-12-12	RJ2206	0	22301056	18	<null>
16	22309180	张三	男	2000-05-15	RJ2201	0	22301056	18	<null>

4 (建立存储过程，在存储过程中输出平均成绩大于 80 分的学生的姓名、性别、年龄和平均成绩。调用该存储过程，并输 4 相应的结果。

```
-- (4) 建立存储过程，在存储过程中输出平均成绩大于80分的学生姓名、性别、年龄和平均成绩。调用该存储过程，并输出相应的结果。
```

```
DELIMITER //
```

```
CREATE PROCEDURE Avg_Grade_Above_80()  
BEGIN  
    SELECT s.Sname, s.Ssex, ROUND(AVG(sc.Grade), 2) AS Avg_Grade  
    FROM student s  
    JOIN sc ON s.Sno = sc.Sno  
    GROUP BY s.Sno, s.Sname, s.Ssex  
    HAVING AVG(sc.Grade) > 80;  
END //
```

```
DELIMITER ;  
CALL Avg_Grade_Above_80();
```

	Sname	Ssex	Avg_Grade
1	葛杨	男	88.00
2	赵陈毅泽	男	100.00
3	王娜	女	96.00
4	杨四敏	女	91.50
5	杨青	女	93.00

console [localhost] student [localhost] sc [localhost] course [localhost]

19 rows

WHERE ORDER BY

	Sno	Cid	Grade
1	22309010	1	69
2	22309100	2	86
3	22309015	2	79
4	22309012	3	88
5	22309048	8	100
6	22309075	8	96
7	22309100	4	70
8	22309148	4	58
9	22309020	4	88
10	22309150	9	93
11	22309160	9	79
12	22301056	1	80
13	22301056	2	80
14	22301056	3	80
15	22309030	3	78
16	22309020	3	95
17	22301056	4	60
18	22301056	5	60
19	22301056	11	60

- 5 写存储过程，根据指定的学生姓名，在过程里输出选修了该学生选修的全部课程的所有学生的学号、姓名、选课数、平均成绩、总学分等信息。调用该存储过程，并根据指定的学生姓名输出与其相应的结果。

```
DELIMITER //
CREATE PROCEDURE Student_Course_Statistics (
    IN p_Sname VARCHAR(8)
)
BEGIN
    -- 声明变量以存储指定学生的学号和选修课程数
    DECLARE v_Sno VARCHAR(8);
    DECLARE v_CourseCount INT;
    -- 根据提供的学生姓名获取学号
    SELECT Sno INTO v_Sno FROM student WHERE Sname = p_Sname;
    -- 计算指定学生选修的课程数
    SELECT COUNT(DISTINCT Cid) INTO v_CourseCount FROM sc WHERE Sno = v_Sno;
    -- 查询选修了与指定学生完全相同课程的其他学生的信息
    SELECT s.Sno, s.Sname, COUNT(sc.Cid) AS Course_Count, ROUND(AVG(sc.Grade), 2) AS Avg_Grade,
        SUM(c.Ccredit) AS Total_Credit
    FROM student s
    JOIN sc ON s.Sno = sc.Sno
    JOIN teaching t ON t.CID=sc.Cid
    JOIN course c ON t.Cno = c.Cno
    WHERE sc.Cid IN (SELECT Cid FROM sc WHERE Sno = v_Sno)
    GROUP BY s.Sno, s.Sname
    HAVING COUNT(DISTINCT sc.Cid) >= v_CourseCount;
END //
DELIMITER ;
CALL Student_Course_Statistics(p_Sname: '杨青');
```

```
DELIMITER ;
CALL Student_Course_Statistics(p_Sname: '杨青');
```

	Sno	Sname	Course_Count	Avg_Grade	Total_Credit
1	22309150	杨青	1	93.00	4
	22309160	杨久玲	1	79.00	4

console [:@localhost] student [:@localhost] sc [:@localhost] x

19 rows v | | | | Tx: Auto v | DDL Q

WHERE ORDER BY

	Sno	Cid	Grade
1	22309010	1	69
2	22309100	2	86
3	22309015	2	79
4	22309012	3	88
5	22309048	8	100
6	22309075	8	96
7	22309100	4	70
8	22309148	4	58
9	22309020	4	88
10	22309150	9	93
11	22309160	9	79
12	22301056	1	80
13	22301056	2	80
14	22301056	3	80
15	22309030	3	78
16	22309020	3	95
17	22301056	4	60
18	22301056	5	60
19	22301056	11	60

- 6 写存储过程显示所有平均成绩小于给定班级号的平均成绩的学生的学号、名字、平均成绩。在过程里输出相关信息。调用该存储过程，并根据指定的班级号输出与其相应的结果。

```
-- (6) 写存储过程显示所有平均成绩小于给定班级号的平均成绩的学生的学号、名字、平均成绩。在过程里输出相关信息。调用该存储过程。
DELIMITER //
CREATE PROCEDURE Avg_Grade_Below_Class (
    IN p_Classno VARCHAR(6)
)
BEGIN
    SELECT s.Sno, s.Sname, ROUND(AVG(sc.Grade), 2) AS Avg_Grade
    FROM student s
    JOIN sc ON s.Sno = sc.Sno
    -- WHERE s.Classno = p_Classno
    GROUP BY s.Sno, s.Sname
    HAVING AVG(sc.Grade) < (
        SELECT AVG(sc.Grade)
        FROM student s
        JOIN sc ON s.Sno = sc.Sno
        WHERE s.Classno = p_Classno
    );
END //
DELIMITER ;
CALL Avg_Grade_Below_Class( p_Classno: 'RJ2201');
```

```
DELIMITER ;
CALL Avg_Grade_Below_Class( p_Classno: 'RJ2201');
```

	Sno	Sname	Avg_Grade
1	22309010	李一在	69.00
2	22309100	田仪	78.00
3	22309015	刘三晶	79.00
4	22309148	赵八砚	58.00
5	22309160	杨久玲	79.00
6	22301056	余旺	70.00
7	22309030	胡五斌	78.00

2. 触发器 (40 分, 每题 5 分)
在 Oracle 下用 PL / SQL 语句定义触发器

7 创建一个当向学生表 Student 中插入一新同学时能自动列出全部同学信息的触发器 Display_Trigger。

执行存储过程 *Insert_Student* , 向学生表中插入一个新同学, 看触发器 *Display_Trigger* 是否被触发。

注意: MySQL 不支持控制台输出, 可将结果插入到新生成的结果表中并展示

```
-- (1) 创建一个当向学生表Student中插入一新同学时能自动列出全部同学信息的触发器Display_Trigger。
-- 执行存储过程Insert_Student, 向学生表中插入一个新同学, 看触发器Display_Trigger是否被触发。
-- 注意, MySQL不支持控制台输出, 可将结果插入到新生成的结果表中并展示。
CREATE TABLE IF NOT EXISTS All_Students (
    Sno VARCHAR(8),
    Sname VARCHAR(8),
    Ssex NVARCHAR(2),
    Sbirthday DATE,
    Classno VARCHAR(6),
    Totalcredit SMALLINT,
    Createdby VARCHAR(8)
);
DELIMITER //
CREATE TRIGGER Display_Trigger AFTER INSERT ON Student
FOR EACH ROW
BEGIN
    -- 插入新同学信息到结果表
    INSERT INTO All_Students (Sno, Sname, Ssex, Sbirthday, Classno, Totalcredit, Createdby)
    VALUES (NEW.Sno, NEW.Sname, NEW.Ssex, NEW.Sbirthday, NEW.Classno, NEW.Totalcredit, NEW.Createdby);
END //
DELIMITER ;

CALL Insert_Student( p_Sno: '22319180', p_Sname: '张四', p_Ssex: '男', p_Sbirthday: '2000-05-15', p_Classno: 'RJ2201');

SELECT * FROM All_Students;
```

Sno	Sname	Ssex	Sbirthday	Classno	Totalcredit	Createdby
22319180	张四	男	2000-05-15	RJ2201	0	22301056

8 创建一个触发器, 当向学生表 Student 中插入一新同学时能自动更新 (增加 1) Class 班级表中该生所在班级的总人数 8 执 3 测试前, 先用 SQL 命令更新 Class 表中各班级人数的数值为实际学生表 Student 中的人数值。

— 由于实际上的学生每个班级没有 20 人, 且每班插入 20 人会使数据量过于庞大, 这里仅计算在 20 的基础上, 有一个 student 加一

1006

1007

1008 ✓

1009

1010

1011

1012

1013

1014

UPDATE class c

SET Studentnumber = (

SELECT COUNT(*)+20

FROM Student s

WHERE s.Classno = c.Classno

);

由于实际上的学生每个班级没有20人，且每班插入20人会使数据量过于庞大，这里仅计算在20的基础上，有一个student加一

WHERE

ORDER BY Studentnumber

	Classno	Classname	Classmajor	Classdept	Studentnumber
1	RJ2204	软件2204	软件工程	智能交通	21
2	RJ2206	软件2206	软件工程	网络安全	21
3	RJ2205	软件2205	软件工程	网络安全	22
4	RJ2202	软件2202	软件工程	软件开发	23
5	RJ2203	软件2203	软件工程	智能交通	23
6	RJ2201	软件2201	软件工程	软件开发	27

DELIMITER //

CREATE TRIGGER Update_Class_Count_Insert AFTER INSERT ON Student

FOR EACH ROW

BEGIN

-- 更新插入学生所在班级的总人数

UPDATE Class

SET Studentnumber = Studentnumber + 1

WHERE Classno = NEW.Classno;

END //

DELIMITER ;

CALL Insert_Student(p_Sno: '22319181', p_Sname: '张五', p_Ssex: '男', p_Sbirthday: '2000-05-15', p_Classno: 'RJ2201');

select * from class;

Result 33-2

work.class

	Classno	Classname	Classmajor	Classdept	Studentnumber
1	RJ2201	软件2201	软件工程	软件开发	28
2	RJ2202	软件2202	软件工程	软件开发	23
3	RJ2203	软件2203	软件工程	智能交通	23
4	RJ2204	软件2204	软件工程	智能交通	21
5	RJ2205	软件2205	软件工程	网络安全	22
6	RJ2206	软件2206	软件工程	网络安全	21

11

9 创建一个触发器，当从学生表 Student 中删除一个同学时能自动更新（减 1）Class 班级表中该生所在班级的总人数。

```
030 -- (3) 创建一个触发器，当从学生表Student中删除一个同学时能自动更新（减1）Class班级表中该生所在班级的总人数。
031 DELIMITER //
032 CREATE TRIGGER Update_Class_Count_Delete AFTER DELETE ON Student
033 FOR EACH ROW
034 BEGIN
035     -- 更新删除学生所在班级的总人数
036     UPDATE Class
037     SET Studentnumber = Studentnumber - 1
038     WHERE Classno = OLD.Classno;
039 END //
040 DELIMITER ;
041 DELETE FROM Student WHERE Sname = '张五';
042 select * from class;
```

	Classno	Classname	Classmajor	Classdept	Studentnumber
1	RJ2201	软件2201	软件工程	软件开发	27
2	RJ2202	软件2202	软件工程	软件开发	23
3	RJ2203	软件2203	软件工程	智能交通	23
4	RJ2204	软件2204	软件工程	智能交通	21
5	RJ2205	软件2205	软件工程	网络安全	22
6	RJ2206	软件2206	软件工程	网络安全	21

- 10 创建一个触发器，当将学生表 Student 中某一个同学从一个班级改为另一个班级时，能自动更新 Class 班级表中该生所在原来班级的总人数（减1）和新班级的总人数（增加1）。

```
-- (4) 创建一个触发器，当将学生表Student中某一个同学从一个班级改为另一个班级时，能自动更新Class班级表中该生所在原来班级的总人数（减1）和新班级的总人数（增加1）。
DELIMITER //
CREATE TRIGGER Update_Class_Count_Update
AFTER UPDATE ON Student
FOR EACH ROW
BEGIN
    -- 原班级总人数减1
    UPDATE Class c1
    SET c1.Studentnumber = c1.Studentnumber - 1
    WHERE c1.Classno = OLD.Classno;

    -- 新班级总人数加1
    UPDATE Class c2
    SET c2.Studentnumber = c2.Studentnumber + 1
    WHERE c2.Classno = NEW.Classno;
END //
DELIMITER ;

UPDATE Student
SET Classno = 'RJ2202'
WHERE Sno = '22309180';

select * from class;
```

```
UPDATE Student
SET Classno = 'RJ2202'
WHERE Sno = '22309180';

select * from class;
```

Classno	Classname	Classmajor	Classdept	Studentnumber
RJ2201	软件2201	软件工程	软件开发	26
RJ2202	软件2202	软件工程	软件开发	24
RJ2203	软件2203	软件工程	智能交通	23
RJ2204	软件2204	软件工程	智能交通	21
RJ2205	软件2205	软件工程	网络安全	22
RJ2206	软件2206	软件工程	网络安全	21

- 11 建一个触发器，当往 Sc 表中插入一个在 Student 表中不存在的学号 Sno 时，就往 Student 表中插入该学号的记录，对于有约束的字段如姓名、性别、班级号等随机产生有效值。

对插入的学生的姓名用学号进行代替，其他都用随机生成

```
0073 DELIMITER //
0074 ✓ CREATE TRIGGER Insert_Student_Record
0075 AFTER INSERT ON Sc
0076 FOR EACH ROW
0077 BEGIN
0078     DECLARE student_count INT;
0079     DECLARE random_class VARCHAR(6);
0080     -- 检查在Student表中是否存在相同的学号
0081     SELECT COUNT(*) INTO student_count FROM Student WHERE Sno = NEW.Sno;
0082
0083     -- 如果学号不存在于Student表中，则插入新的学生记录
0084     IF student_count = 0 THEN
0085         -- 从Class表中随机选择一个班级
0086         SELECT Classno INTO random_class FROM Class ORDER BY RAND() LIMIT 1;
0087
0088         -- 插入新的学生记录
0089         INSERT INTO Student (Sno, Sname, Ssex, Sbirthday, Classno)
0090         VALUES (NEW.Sno, NEW.Sno, IF(RAND() > 0.5, '男', '女'), NOW(), random_class);
0091     END IF;
0092 END //
0093 DELIMITER ;
0094
0095 ✓ INSERT INTO Sc (Sno, Cid, Grade)
0096 VALUES ('22309123', '1', 85);
0097 ✓ select * from student;
```

Sno	Sname	Ssex	Sbirthday	Classno	Totalcredit	Createdby	Age	
11	22309088	秦琪健	男	2005-03-01	RJ2203	0	22301056	18
12	22309100	田仪	女	2006-02-26	RJ2204	0	22301056	18
13	22309123	22309123	女	2024-04-23	RJ2201	0	22301056	18
14	22309148	赵八砚	男	2005-04-25	RJ2205	0	22301056	18
15	22309150	杨青	女	2005-11-15	RJ2205	0	22301056	18
16	22309160	杨久玲	女	2005-12-12	RJ2206	0	22301056	18
17	22309180	张三	男	2000-05-15	RJ2202	0	22301056	18
18	22319180	张四	男	2000-05-15	RJ2201	0	22301056	18

12 写一个触发器阻止将学生成绩降低。

```
1102 -- (6) 写一个触发器阻止将学生成绩降低。
1103 DELIMITER //
1104 ✓ CREATE TRIGGER Prevent_Grade_Decrease
1105 BEFORE UPDATE ON Sc
1106 FOR EACH ROW
1107 BEGIN
1108     DECLARE old_grade INT;
1109
1110     -- 获取旧成绩
1111     SELECT Grade INTO old_grade FROM Sc WHERE Sno = NEW.Sno AND Cid = NEW.Cid;
1112
1113     -- 如果新成绩比旧成绩低，则阻止更新
1114     IF NEW.Grade < old_grade THEN
1115         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = '不允许将成绩降低';
1116     END IF;
1117 END //
1118 DELIMITER ;
1119
1120 ❶ UPDATE sc
1121 SET grade = grade-1
1122 WHERE Sno = '22301056' and Cid=1;
1123
1124 select * from sc;
1125
1126
```

[45000][1644] 不允许将成绩降低 Stop Retry Ignore

13 在 Sc 表上创建触发器，只要有人选修的课程超过 3 门，就中断操作并提示警告。使用本人学号进行测试，检查触发器的执行并输出结果表的内容。

下表可以看到本人学号的可成已经超过 3 门了，所以执行插入操作会提示并且中断操作

```
1127
1130 DELIMITER //
1131 ✓ CREATE TRIGGER Check_Course_Count
1132 BEFORE INSERT ON sc
1133 FOR EACH ROW
1134 BEGIN
1135     DECLARE course_count INT;
1136     -- 计算学生选修的课程数量
1137     SELECT COUNT(*) INTO course_count FROM Sc WHERE Sno = NEW.Sno;
1138
1139     -- 如果课程数量超过3门，则中断操作并提示警告
1140     IF course_count > 3 THEN
1141         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = '选修的课程数量超过了3门，请注意!';
1142     END IF;
1143 END //
1144 DELIMITER ;
1145 ❶ INSERT INTO Sc (Sno, Cid, Grade) VALUES ('22301056', 8, 85);
1146
1147
1148
```

[45000][1644] 选修的课程数量超过了3门，请注意!

	Sno	Cid	Grade
1	22301056	1	80
2	22301056	2	80
3	22301056	3	80
4	22301056	4	60
5	22301056	5	60
6	22301056	11	60
7	22309010	1	69
8	22309012	3	88
9	22309015	2	79
10	22309020	4	88
11	22309020	3	95
12	22309030	3	78
13	22309048	8	100
14	22309075	8	96
15	22309100	2	86
16	22309100	4	70
17	22309123	1	85
18	22309148	4	58
19	22309150	9	93
20	22309160	9	79

- 14 创建一个触发器，当往 Sc 表插入选修的课程时自动将该课程的学分累加到 Student 表的该生的总学分 (Totalcredit) 中，当从 Sc 表中退课时自动从 Student 表的该生的总学分中减去该课程的学分，最终学分数不能小于 0。测试执行删除原来的所有记录，然后重新执行数据的插入，检查触发器的执行并输出结果表的内容。

可以看到删除所有记录后学分为 0，新插入的三个数据的学分分别为 4,4,4，总和为 12


```
156 DROP TRIGGER IF EXISTS Update_Total_Credit_Insert;
157 -- 创建触发器，在Sc表中插入选修的课程时自动将该课程的学分累加到Student表的该生的总学分（Totalcredit）中
157 DELIMITER //
158 CREATE TRIGGER Update_Total_Credit_Insert
159 AFTER INSERT ON Sc
160 FOR EACH ROW
161 BEGIN
162     -- 更新学生的总学分
163     UPDATE Student
164     SET Totalcredit = Totalcredit + (SELECT Ccredit FROM Course WHERE Cno =(select cno from teaching
165                                     where teaching.CID= NEW.Cid) )
166     WHERE Sno = NEW.Sno;
167 END //
168
169 CREATE TRIGGER Update_Total_Credit_Delete
170 AFTER DELETE ON Sc
171 FOR EACH ROW
172 BEGIN
173     -- 更新学生的总学分，但最终学分不能小于0
174     UPDATE Student
175     SET Totalcredit = GREATEST(0, Totalcredit - (SELECT Ccredit FROM Course WHERE Cno =(select cno from teaching
176                                     where teaching.CID= OLD.Cid) ))
177     WHERE Sno = OLD.Sno;
178 END //
179 DELIMITER ;
180
```

```
1181 -- 删除原来的所有记录
1182 DELETE FROM Sc;
1183
1184 -- 重新执行数据的插入
1185 INSERT INTO Sc (Sno, Cid, Grade) VALUES
1186 ('22301056', 1, 80),
1187 ('22301056', 2, 85),
1188 ('22301056', 3, 90);
1189 select * from student;
```

	Sname	Ssex	Sbirthday	Classno	Totalcredit	Createdby	Age	Addr
1	余旺	男	2004-04-26	RJ2202	12	22301056	19	<null>
2	李一在	男	2005-10-01	RJ2201	0	22301056	18	<null>
3	葛杨	男	2006-08-08	RJ2201	0	22301056	18	<null>
4	刘三晶	女	2005-05-22	RJ2201	0	22301056	18	<null>
5	杨四敏	女	2007-01-08	RJ2201	0	22301056	18	<null>
6	胡五斌	男	2006-10-08	RJ2201	0	22301056	18	<null>
7	赵陈毅泽	男	2005-06-06	RJ2202	0	22301056	18	<null>
8	王威	男	2005-06-10	RJ2202	0	22301056	18	<null>