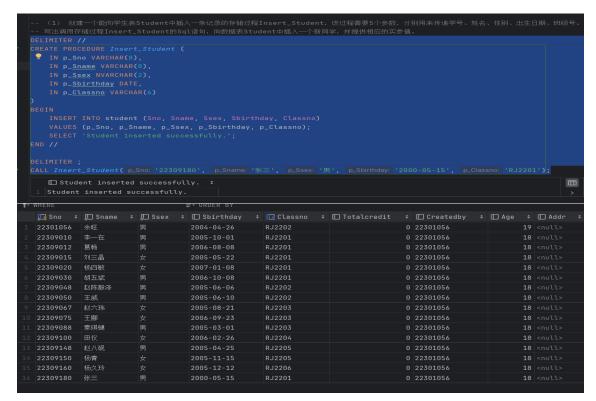
work-sql

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。



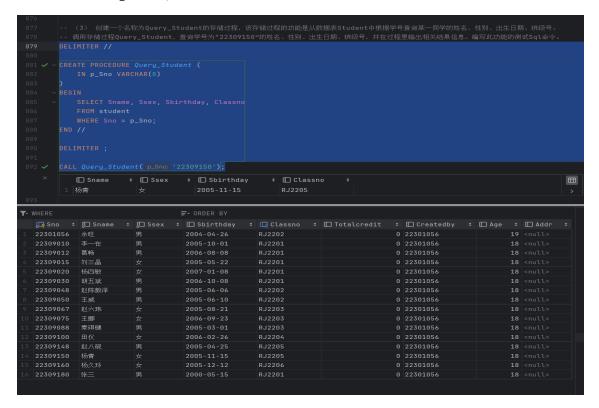
2 创建一个向课程表 Course 中插入一门新课程的存储过程 Insert_Course,该存储过程需要三个参数,分别用来传递课程号、课程名、学分,但允许参数"学分"的默认值为 4,即当调用存储过程 I 时,未给第三个参数"学分"提供实参值时,存储过程将按默认值 4 进行运算。

调用存储过程 Insert_Course,向课程表 Course 中插人一门新课程。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。



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。

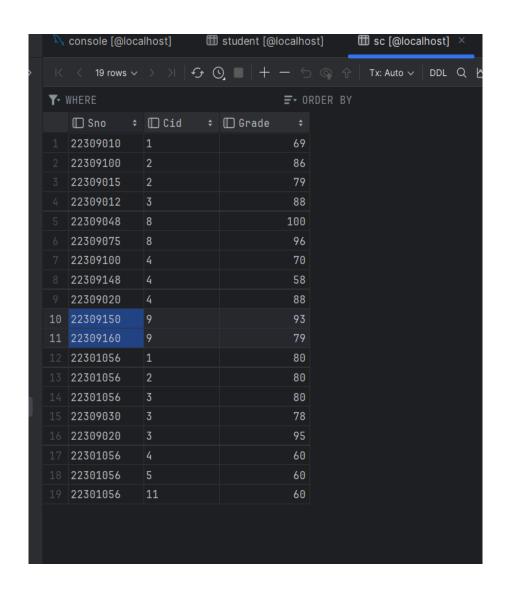


4 (建立存储过程,在存储过程中输出平均成绩大于 80 分的学生的姓名、性别、年龄和平均成绩。调用 该存储过程,并输 4 相应的结果。

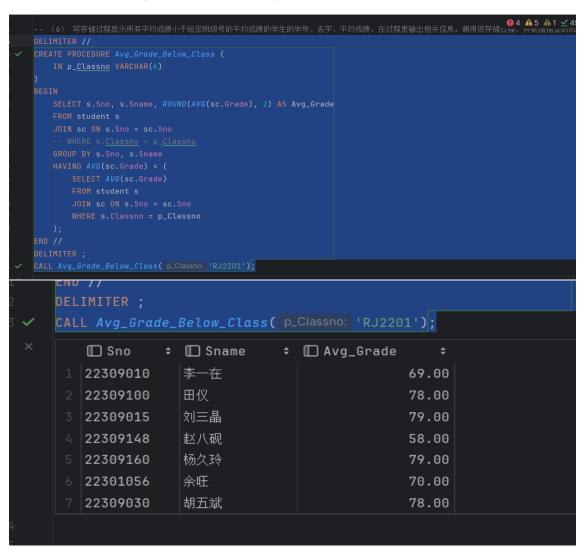
95		体性和支持山东均县			* 计和 一		
96 97	(4) 建立存储过程,在存 DELIMITER //	储过程中制出半30.0	《赖天丁887的学生的姓名、性	31、牛龄和平均成绩。调用该仔1	和过程,开输出:相应的结果。 ————————————————————————————————————		
98							
99 ~ 90	✓ CREATE PROCEDURE Avg_Grade_Above_80() BEGIN SELECT s.Sname, s.Ssex, ROUND(AVG(sc.Grade), 2) AS Avg_Grade						
91							
92	FROM student s JOIN sc ON s.Sno = sc.Sno GROUP BY s.Sno, s.Sname, s.Ssex						
93 94							
95	HAVING AVG(sc.Grade						
96 97							
98	DELIMITER ;						
99 🗸	CALL Avg_Grade_Above_80();						
×	☐ Sname ÷ ☐ Ss	ex ÷ 🗅 Avg					
	1 葛畅 男 2 赵陈毅泽 男		88.00 100.00				
	3 王娜 女		96.00				
	4 杨四敏 女		91.50				
10	5 杨青 女		93.00				
11							
	🔪 console [@loca	alhost] fff	student [@localhost]	⊞ sc [@localhost] ×	⊞ course [@loc		
	* { CONSOIC [@IOCK	an103tj w	student [@iocainost]	m sc [@iocainost]			
	<	ightarrow $ ightarrow$ $ ightarrow$ $ ightarrow$ $ ightarrow$	9 = + - 5 @	⊕ Tx: Auto ∨ DDL C	. <u>W</u>		
	Y → WHERE		≡ - ORDER	RV			
		Поза.					
			☐ 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。

6	DELIMITER //						
7 🗸 🗸	CREATE PROCEDURE Student_Course_Statistics (
8	IN p_Sname_VARCHAR(8)						
9							
0 ~	BEGIN						
1	声明变量以存储指定学生的学号和选修课程数						
2	DECLARE v_Sno VARCHAR(8);						
3	DECLARE v_CourseCount INT;						
4							
5	SELECT Sno INTO v_Sno FROM student WHERE Sname = p_Sname;						
6	计算指定学生选修的课程数						
7	SELECT COUNT(DISTINCT Cid) INTO v_CourseCount FROM sc WHERE Sno = v_Sno;						
8	查询选修了与指定学生完全相同课程的其他学生的信息						
9 ~	SELECT s.Sno, s.Sname, COUNT(sc.Cid) AS Course_Count, ROUND(AVG(sc.Grade), 2) AS Avg_Grade,						
0	SUM(c.Ccredit) AS Total_Credit						
1	FROM student s						
2	JOIN sc ON s.Sno = sc.Sno						
3	JOIN teaching t on t.CID=sc.Cid						
4	JOIN course c ON t.Cno = c.Cno						
,	WHERE sc.Cid IN (SELECT Cid FROM sc WHERE Sno = v_Sno)						
0	GROUP BY s.Sno, s.Sname						
0	HAVING COUNT(DISTINCT sc.Cid) >= v_CourseCount;						
0	END //						
7 O	DELIMITER;						
	CALL Student_Course_Statistics(p_Sname: '杨青');						
in in	DELIMITER;						
11							
10 V	CALL Student_Course_Statistics(p_Sname: '杨青');						
12							
X	☐ Sno						
	1 22309150 杨青 1 93.00 4						
	22309160 杨久玲 1 79.00 4						
13							



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。

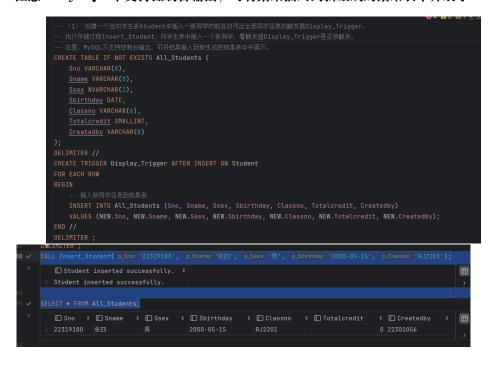


2. 触发器 (40 分, 每题 5 分 在 Oracle 下用 PL / SQL 语句定义触发器

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。

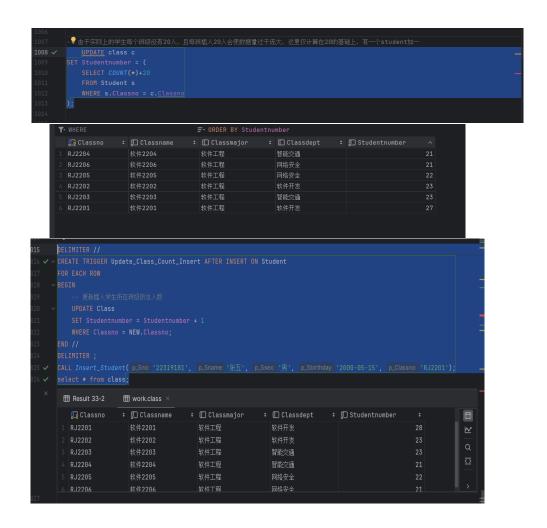
执行存储过程 Insert_Student ,向学生表中插入一个新同学,看触发器 Display_Trigger 是否被触发。

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

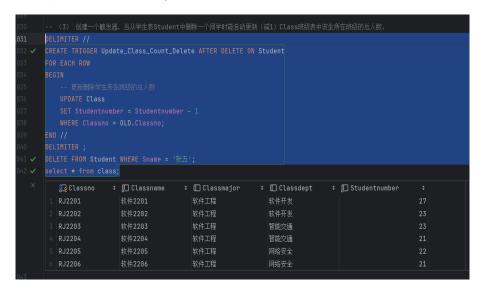


8 创建一个触发器,当向学生表 Student 中插入一新同学时能自动更新(增加 1) Class 班级表中该生所在班级的总人数 8 执 3 测试前,先用 SQL 命令更新 Class 表中各班级人数的数值为实际学生表 Student 中的人数值。

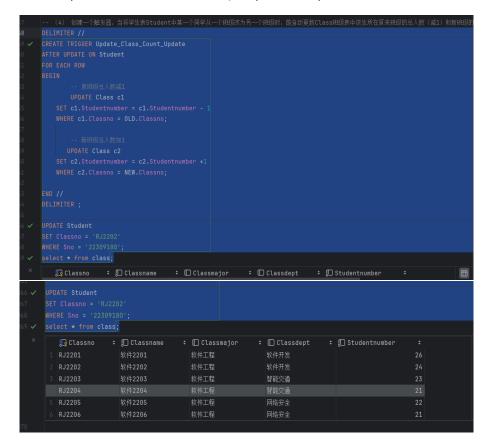
- 由于实际上的学生每个班级没有 20 人,且每班插入 20 人会使数据量过于庞大,这里仅计算在 20 的基础上,有一个 student 加一



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。



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)。



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

```
### PRINTER //
### CREATE TRISER TO SC
### CREATE TRISER TRISER TO SC
### CREATE TRISER TRISER TRISER TRISER
### CREATE TRISER TRISER TRISER TRISER
### CREATE TRISER TRISER TRISER TRISER
### CREATE TRISER TRISER
```

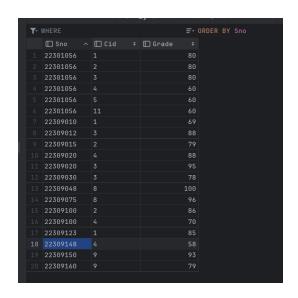
12 写一个触发器阻止将学生成绩降低。

```
1102 -- (6) 写一个触发器阻止将学生成绩降低.

1103 DELIMITER //
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 -- 如果新成绩比旧成绩低、则阻止更新
114 IF NEW.Grade < old_grade THEN
115 SIGNAL SQLSTATE '45080' SET MESSAGE_TEXT = '不允许将成绩降低';
116 END IF;
117 END //
118 DELIMITER;
119
1109 UPDATE sc
1111 Signade = grade-1
1122 WHERE Sno = '22381956' and Cid=4';
1123
1124 Select * from sc;
```

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



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