

数据库系统课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 触发器 |
| 实验日期： | 2025.5.9 |
| 实验地点： | 西部片区4号楼104 |
| 提交日期： | 2025.5.16 |
|  | |
| 学号： | 37220232203786 |
| 姓名： | 潘腾凯 |
| 专业年级： | 软工2023级 |
| 学年学期： | 2024-2025学年第二学期 |

1. 实验目的

• 理解和掌握 Mysql8.4 触发器的作用和工作原理

■ AFTER/BEFORE 触发器

■ 行级触发器

• 熟练掌握 Mysql8.4 触发器的设计方法

• 熟练掌握 Mysql8.4 触发器创建、 删除与查看的方法

1. 实验内容和步骤

1.关于 mysql8.4 触发器的两点知识：

（1） mysql8.4 只支持行级触发器， 不支持语句级触发器

（2） mysql8.4 使用修饰词 old 和 new 来代表修改前后的值，具体使用如下：

|  |  |  |
| --- | --- | --- |
| Trigger Event | OLD | NEW |
| INSERT | No | Yes |
| UPDATE | Yes | Yes |
| DELETE | Yes | No |

2.Before 触发器的创建与验证

创建触发器的语法

CREATE TRIGGER trigger\_name

{BEFORE | AFTER} {INSERT | UPDATE | DELETE} ON table\_name

FOR EACH ROW

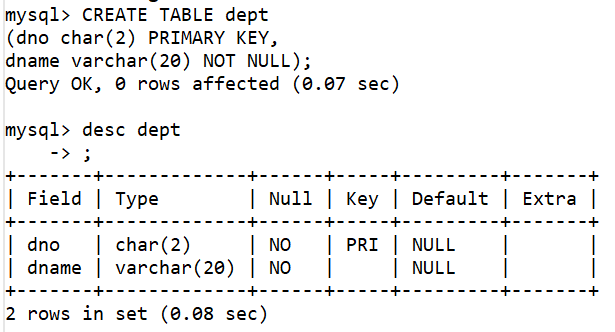
BEGIN

-- Trigger body (SQL statements)

END; \g --触发器代码最后以\g 结尾，表示执行代码，如要中断执行，同时按 CTRL+C

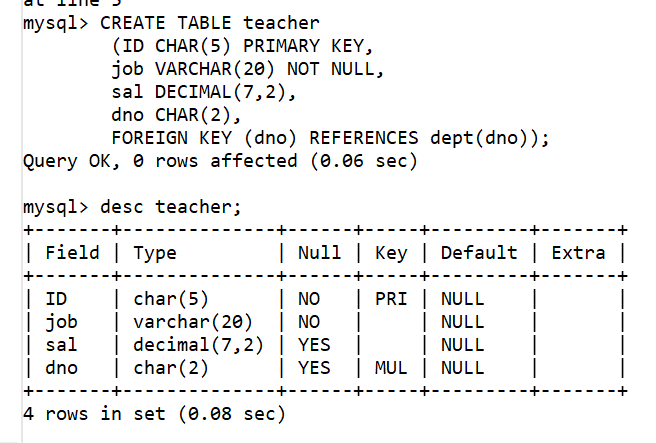
1. 创建部门表 dept(dno, dname)，其中， dno 为部门号， 定长为 2 的字符型，主码； dname 为部门名， 最大长度为 20 的变长字符型，非空

如下：



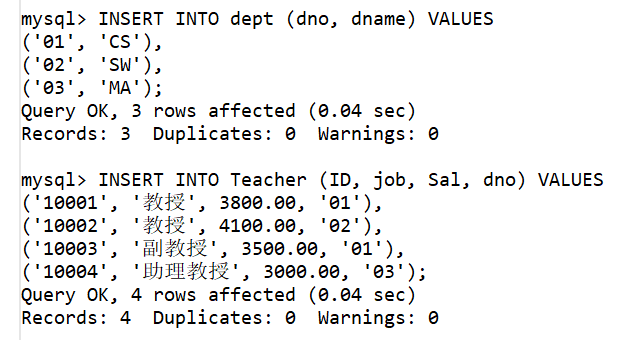
1. 创建 Teacher 表： Teacher(ID, job, Sal, dno)，其中， ID 为教工号，定长为 5 的字符型，主码； job 为职称，最大长度为 20 的变长字符型，非空； sal 为工资， 总长度为 7 的数值型，其中保留两位小数； dno 为部门号，定长为 2 的字符型，外码，引用 dept 表中的主码 dno

如下：



1. 为dept表增加实验数据： (‘01’,’CS’), (‘02’,’SW’), (‘03’,’MA’)； 为Teacher表增加实验数据： (‘10001’,’教 授 ’,3800,’01’), (‘10002’,’ 教 授 ’,4100,’02’), (‘10003’,’ 副 教 授 ’,3500,’01’), (‘10004’,’ 助 理 教授’,3000,’03’)

如下：



（4） 在 Teacher 表上创建一个名为 tri\_INSERT\_SAL 的 BEFORE 行级触发器以实现如下完整性规则： 如

果新增数据中教授的工资低于 4000 元，则自动改为 4000 元

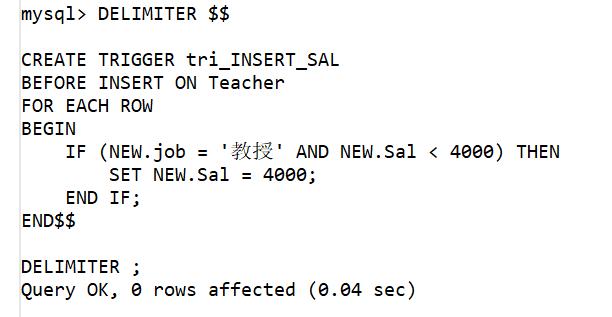
注： 本步骤使用了 IF 语句， trigger body 的格式：

if（condition） then set new.sal =4000; --设置字段的值使用 SET…=;

end if;

--格式已经给出， 主要工作是将 condition 替换为具体的表达式

如下：

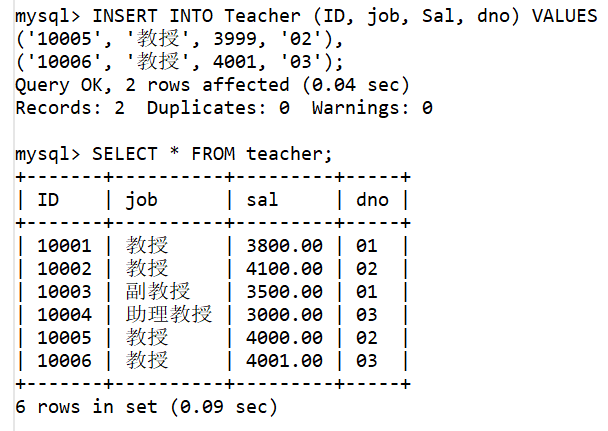


因为触发器体内部使用分号（‘;‘），所以用DELIMITER临时改变MySQL的语句结束符为’$$’来避免冲突。

（5） 验证 tri\_INSERT\_SAL 触发器是否正常工作：

A.插入两条新数据(‘10005’,’教授’,3999,’02’)， (‘10006’,’教授’,4001,’03’)

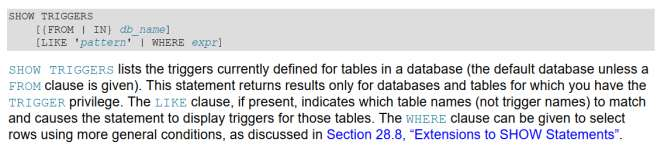
B.发布查询语句： select \* from teacher; 观察触发器是否按预期执行成功



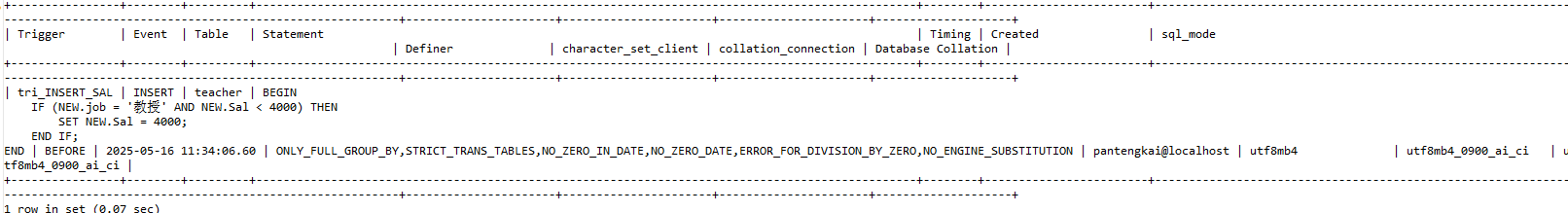
触发器按预期执行成功

3.查看触发器

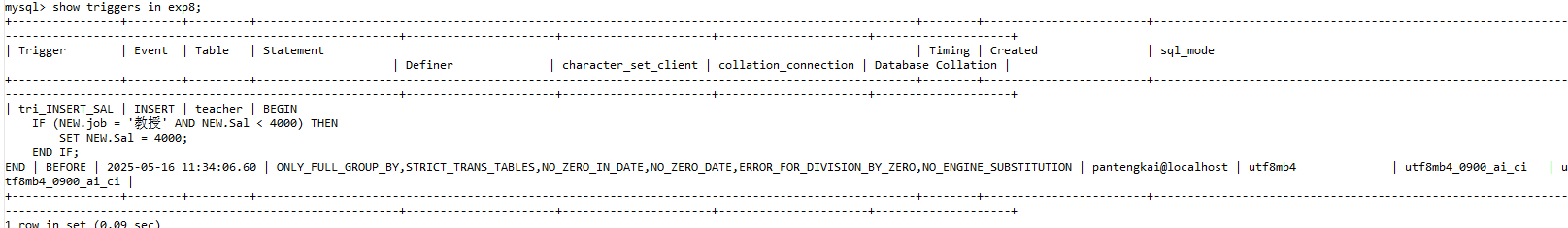
命令： show triggers…\g



• show triggers\g ---显示当前数据库的所有触发器信息，包括名称和代码

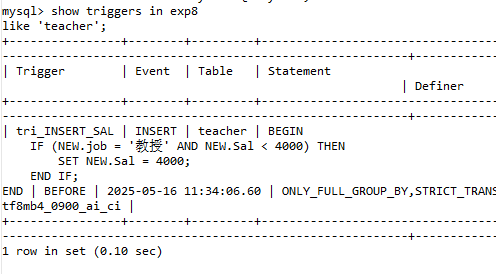


• show triggers [from | in] db\_name \g ---显示 db\_name 数据库里的触发器信息



• like ‘pattern’ --这里的 pattern 是指表名， 非触发器名

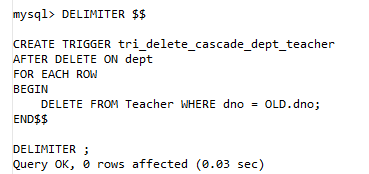
特别注意：执行详细用法见： mysql8.4 手册的第 15 章之 15.7.7.40 SHOW TRIGGERS Statement



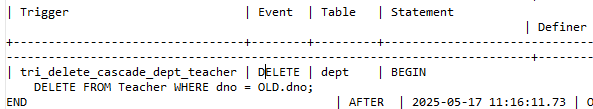
4.触发器实现外码约束

（1） 设计触发器自动维持以下表间的外码约束： 删除 dept 表中 dno 的数据（如 dno=’03’） 后， teacher 表上引用该数据的记录也被自动删除。

如下：



Trigger成功创建：



（2）验证（1）的效果。发布语句：

delete from dept where dno=’03’;\g

select \* from teacher;\g --观察 dno=’03’的记录是否被删除， 从而验证触发器是否正常工作

需要注意的是，如果已定义了 teacher.dno 引用 dept.dno（即外码），那么触发器虽然工作，但因有外码约束，所以删除 teacher 表的相应数据失败（系统将会提示）

删除失败：



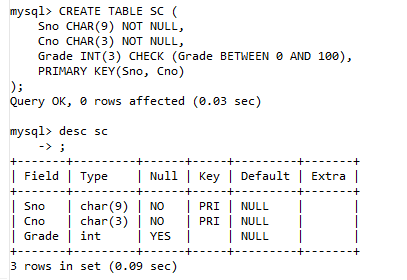
5.触发器实现审计功能

设有选课表 SC(Sno,Cno,grade)： sno： 学号， 定长为 9 的字符型，非空； cno： 课程号， 定长为 3 的字符型，非空； grade： 课程成绩， 长度为 3 的整型，约束：介于 0 和 100 之间

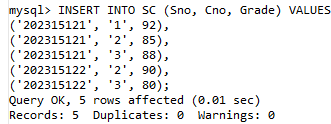
样本数据集为： {(‘202315121’,’1’,92), (‘202315121’,’2’,85), (‘202315121’,’3’,88), (‘202315122’,’2’,90), (‘202315122’,’3’,80)}

要求设计一个触发器实现审计日志记录：当对表 SC 的 Grade 属性进行修改时，若分数增加了 10%及其以上，则将此次操作登记到下表中： SC\_U(Sno, Cno, Oldgrade, Newgrade)，其中各字段含义和数据类型： sno 同 SC.sno， cno 同 SC.cno， 修改前成绩 oldgrade 和修改后成绩 newgrade 同 SC.grade。操作步骤如下：

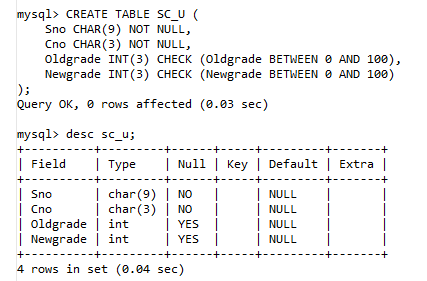
1. 创建 SC 表： SC(sno, cno, grade)



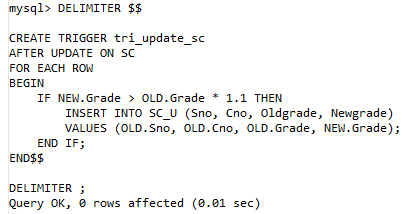
（2） 插入样本数据



（3） 创建 SC\_U 表： SC\_U(Sno, Cno, Oldgrade, Newgrade)



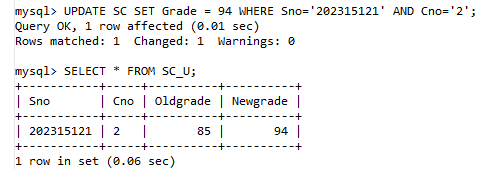
（4） 在 SC 表上创建一个名为 tri\_update\_sc 的 AFTER 行级触发器



（5） 验证 tri\_update\_sc 触发器是否正常工作？

A. update sc set grade =94 where sno='202315121' and cno='2';

B. select \* from sc\_u;



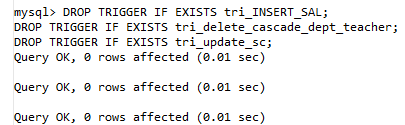
正常工作

6.删除触发器

删除命令： DROP TRIGGER [IF EXISTS] [schema\_name.]trigger\_name;

• 删除所有的触发器

• 注意： 如果删除触发器所在的基本表，那么触发器将被自动删除而无需显式删除



1. 实验总结

3.1 完成的工作

完成了mysql84各种功能（插入，删除，更新）触发器的创建、验证和删除。

3.2 对实验的认识

1.认识：

深入理解触发器的工作机制：触发器是响应特定事件（如 INSERT, UPDATE, DELETE）而自动执行的存储过程。实验中创建了不同类型的触发器（如 BEFORE INSERT, AFTER DELETE, AFTER UPDATE），以满足不同的业务规则需求。

学会了如何编写触发器：包括条件判断（如 IF 语句）、对新旧数据的访问（如 NEW 和 OLD 关键字）等技巧。

学会了如何使用触发器维护数据的一致性：例如，在更新成绩时如果分数增加了超过10%，则将更改前后的信息记录到审计表中；或者确保教授的工资不低于某个标准值。

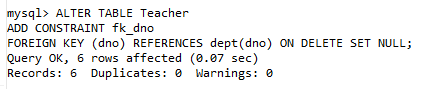
增强了对数据安全性的认识：通过触发器可以在一定程度上防止不合法的数据修改或删除操作，从而保护数据库的安全性和完整性。

2.思考题  
2.1如何实现在删除 dept 表中 dno 为’01’的数据后自动将 teacher 表中引用了该值（即’01’） 的 dno 字段值设置为 null？请验证。

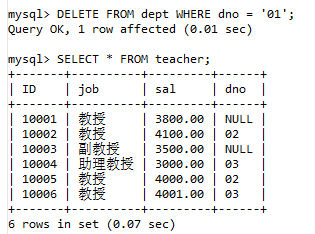
先删除原有外键约束：



再添加on delete set null的约束



验证：



3.3 遇到的困难及解决方法

无。