|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 《数据库原理》实验报告 | | | | |
| 题目：实验五  存储过程，触发器与存储函数 | 学号 | 姓名 | 班级 | 日期 |
| **2021302111** | **禹泽海** | **JKC012101** | **2023/11/4** |

**一 .头歌平台截图**



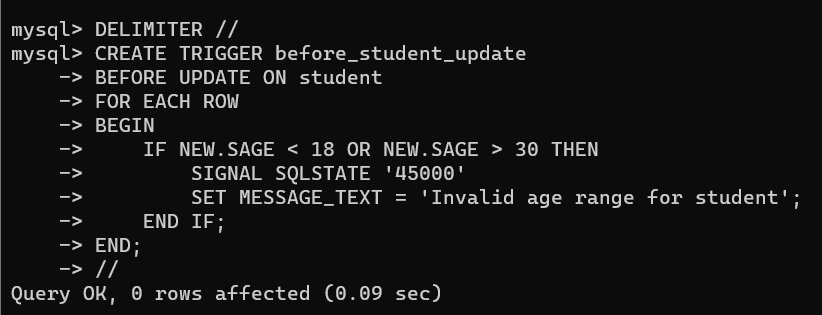
头歌平台实验五完成截图

**二 .实验内容、步骤以及结果（线下实验）**

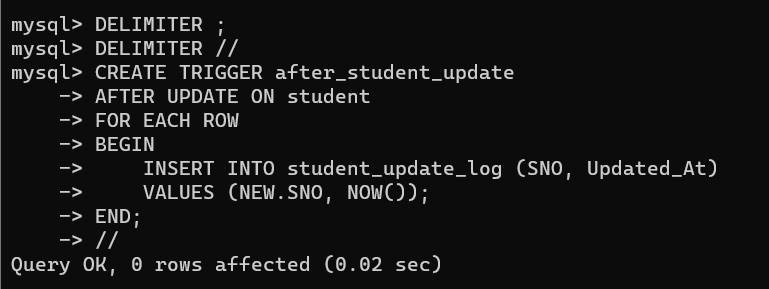
**题目：自己设计一个before update触发器和after update触发器，比较before 和after触发器的区别**

**1 .首先在student数据库中定义两类触发器**

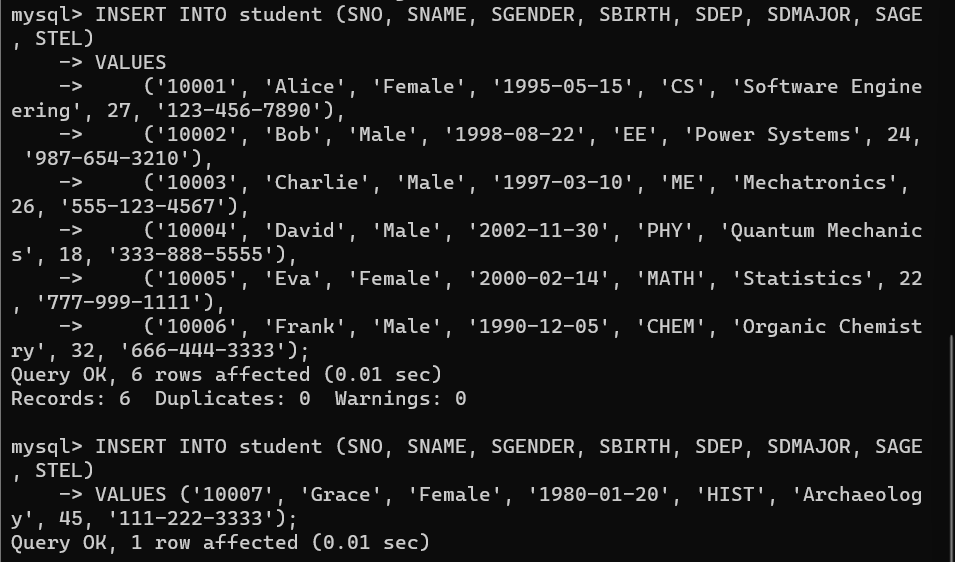
**Before触发器：**

****

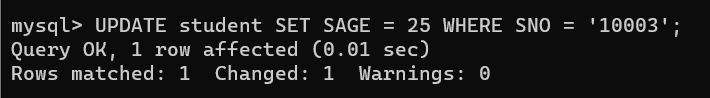
**After触发器：**

****

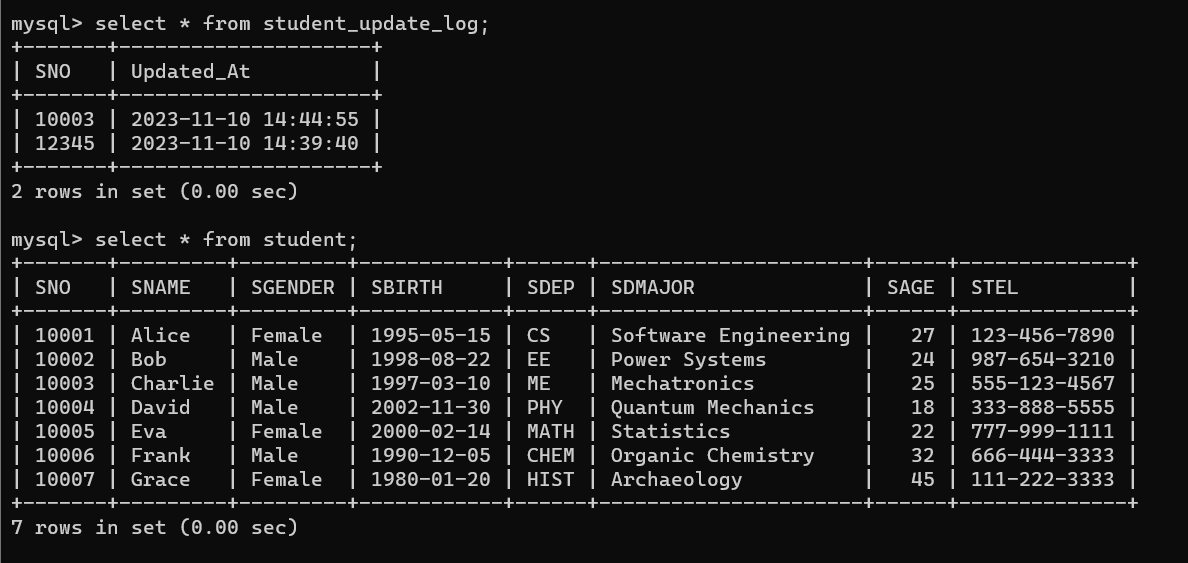
**2 .插入数据（先前实验插入10w数据过于庞大，先进行清空表处理）：**

****

**3 .使用更新语句检验触发器：**

****

**4 .查询更新之后的两个表对结果进行分析：**

****

**可以看到触发器效果得到了验证：通过更新学生表的记录，触发了名为after\_student\_update的after update触发器。该触发器将更新的学号（SNO）和更新的时间（Updated\_At）插入到student\_update\_log表中。从student\_update\_log表的查询结果可以看到，学号为'10003'的记录在2023-11-10 14:44:55被成功更新，而学号为'12345'的记录在2023-11-10 14:39:40被更新。这验证了after update触发器的有效性。**

**同时学生表的更新效果也可以看出，通过UPDATE语句将学号为'10003'的学生年龄更新为25岁，成功影响了一行记录。从student表的查询结果可以看到，学号为'10003'的学生的年龄确实被更新为25岁。**

**通过这个实验，我们深入理解了MySQL数据库中的数据插入、触发器的创建与验证，并对触发器的不同类型（before和after）进行了比较。这有助于我们更好地利用数据库的功能来确保数据的完整性、一致性和准确性。在实际应用中，触发器可以用于实现业务规则、审计功能等，提升数据库的管理和操作水平。**

**三 .实验中出现的问题以及解决方案**

**问题与解决方案：**

**在第一次尝试插入数据时，出现了错误 1406 (22001) - "Data too long for column 'SDEP' at row 2"。这是因为SDEP字段的长度设置为varchar(20)，而在第二行插入时，'Computer Science'超过了这个长度。在第二次尝试时，成功插入了6条数据，并且在第三次尝试时，也成功插入了一条数据。这表明对数据类型和字段长度的设置需要仔细考虑，以避免数据插入错误。**

**思考感悟：**

**当学习MySQL的存储过程、触发器和存储函数时，我深刻体会到这些数据库编程元素的重要性和灵活性。通过存储过程，我能够将一系列SQL语句封装成一个可重复使用的单元，提高了代码的模块性和可维护性。这对于复杂的业务逻辑尤其有益，使得数据库操作更为高效。**

**触发器的学习让我意识到在数据库中，事件的发生可以触发自动的响应。这为数据的一致性和完整性提供了保障，尤其在需要对关联表进行同步操作时，触发器能够确保相关的数据变更都得到正确处理。这是一种强大的机制，让数据库管理更为自动化。**

**此外，存储函数的运用使得我能够在查询中嵌入自定义的业务逻辑，使得数据库操作更为灵活。通过存储函数，我可以将复杂的计算过程封装起来，简化了查询语句，提高了代码的可读性和可维护性。总体而言，这些数据库编程元素为我提供了更为高效和精细的数据管理工具，让我更好地应对实际应用中的各种需求。**

**批阅者：**

**批阅日期：**

**实验成绩：**

**批注：**