**暨南大学本科实验报告专用纸**

课程名称 《数据库系统》 成绩评定

实验项目名称 指导教师雷小林 、魏林锋

实验项目编号 实验项目类型 综合 实验地点 N117

学生姓名 金鸿正 学号 2021102560

学院 信息科学与技术学院 系 计算机 专业 软件工程

实验时间 年 月 日 午～ 月 日 午 温度 ℃湿度

4 触发器实验

4.1 实验目的

掌握数据库触发器的设计和使用方法。

4.2 实验内容和要求

定义 BEFORE 触发器和 AFTER 触发器。能够理解不同类型触发器的作用和执行原理，验证触发器的有效性。

4.3 实验重点和难点

实验重点：触发器的定义。

实验难点：利用触发器实现较为复杂的用户自定义完整性。

4.4 实验内容记录

4.4.1 AFTER 触发器

AFTER 触发器可以用来维护一致性。

比如有一张员工总工资表，那么在加入新的工资记录后，需要更新总工资表。

总工资表：

CREATE TABLE emp\_salary\_total (

emp\_no INT NOT NULL,

salary\_total INT NOT NULL DEFAULT 0,

CONSTRAINT FK\_empno FOREIGN KEY (emp\_no)

REFERENCES employees(emp\_no)

);

建立总工资表后，初始化一次数据：

INSERT INTO emp\_salary\_total

SELECT emp\_no, SUM(salary)

FROM salaries

GROUP BY emp\_no;

在总工资表上定义触发器：

DELIMITER $$

CREATE TRIGGER TRI\_salary\_total\_DELETE

AFTER DELETE ON salaries

FOR EACH ROW

BEGIN

UPDATE emp\_salary\_total e

SET e.salary\_total = (

SELECT SUM(salary)

FROM salaries s

WHERE s.emp\_no=OLD.emp\_no

)

WHERE e.emp\_no=OLD.emp\_no;

END $$

下面进行验证：

先查询 10001 的总工资：

SELECT \*

FROM emp\_salary\_total

WHERE emp\_no=10001 $$

得到总工资为：

+--------+--------------+

| emp\_no | salary\_total |

+--------+--------------+

| 10001 | 1281612 |

+--------+--------------+

MariaDB [employees]> select \* from salaries where emp\_no=10001$$

+--------+--------+------------+------------+

| emp\_no | salary | from\_date | to\_date |

+--------+--------+------------+------------+

| 10001 | 60117 | 1986-06-26 | 1987-06-26 |

| 10001 | 62102 | 1987-06-26 | 1988-06-25 |

| 10001 | 66074 | 1988-06-25 | 1989-06-25 |

| 10001 | 66596 | 1989-06-25 | 1990-06-25 |

| 10001 | 66961 | 1990-06-25 | 1991-06-25 |

| 10001 | 71046 | 1991-06-25 | 1992-06-24 |

| 10001 | 74333 | 1992-06-24 | 1993-06-24 |

| 10001 | 75286 | 1993-06-24 | 1994-06-24 |

| 10001 | 75994 | 1994-06-24 | 1995-06-24 |

| 10001 | 76884 | 1995-06-24 | 1996-06-23 |

| 10001 | 80013 | 1996-06-23 | 1997-06-23 |

| 10001 | 81025 | 1997-06-23 | 1998-06-23 |

| 10001 | 81097 | 1998-06-23 | 1999-06-23 |

| 10001 | 84917 | 1999-06-23 | 2000-06-22 |

| 10001 | 85112 | 2000-06-22 | 2001-06-22 |

| 10001 | 85097 | 2001-06-22 | 2002-06-22 |

| 10001 | 88958 | 2002-06-22 | 9999-01-01 |

+--------+--------+------------+------------+

17 rows in set (0.00 sec)

从中删除一条数据：

DELETE FROM salaries

WHERE salary=60117

AND emp\_no=10001 $$

再查询总工资为：

+--------+--------------+

| emp\_no | salary\_total |

+--------+--------------+

| 10001 | 1221495 |

+--------+--------------+

符合预期。

在触发器上不应定义过于复杂（耗时）的动作，例如本例中触发器执行了全扫描更新。更好的处理方式是利用被删除的行的信息来增量更新总工资。

DELIMITER $$

CREATE TRIGGER TRI\_salary\_total\_DELETE

AFTER DELETE ON salaries

FOR EACH ROW

BEGIN

UPDATE emp\_salary\_total e

SET e.salary\_total = e.salary\_total - OLD.salary

WHERE e.emp\_no=OLD.emp\_no;

END $$

此时再删除一条数据，

DELETE FROM salaries

WHERE salary=62102

AND emp\_no=10001 $$

然后查询总工资为

+--------+--------------+

| emp\_no | salary\_total |

+--------+--------------+

| 10001 | 1159393 |

+--------+--------------+

同样符合预期。

4.4.2 BEFORE 触发器

BEFORE 触发器可以用来检查数据更新的合法性，可以用来实现比断言更为复杂的检查（断言只能定义一个 CHECK 子句）。

例如，加入一个日期检查，要求 to\_date 必须大于 from\_date 。

注：MYSQL 抛出异常的写法和书中给出的不同。

DELIMITER $$

CREATE TRIGGER TRI\_salaries\_INSERT

BEFORE INSERT ON salaries

FOR EACH ROW

BEGIN

DECLARE

v\_msg VARCHAR(200);

IF (NEW.from\_date >= NEW.to\_date)

THEN

BEGIN

SET v\_msg = 'to\_date is EARLIER than from\_date!';

SIGNAL SQLSTATE 'HY000' SET MESSAGE\_TEXT = v\_msg;

END;

END IF;

END; $$

然后插入一条一场数据：

INSERT INTO salaries

VALUES (10001, 15000, '2020-03-01', '2020-01-01');

会得到错误：

ERROR 1644 (HY000): to\_date is EARLIER than from\_date!

**暨南大学本科实验报告专用纸(附页)**