

# 数据库实验报告 实验十二 触发器

姓名	学号	班级	课室
熊明	20305055	计科5班	D503

## 一、实验目的

通过实验使学生加深对数据完整性的理解，学会创建和使用触发器

## 二、实验环境

数据库：Mysql

图形化工具：Navicat Premium 16

## 三、实验内容

PostgreSQL 触发器是一种特殊的函数，当指定的数据库事件发生时自动执行/调用。触发器可以在数据插入、更新或删除时自动运行一段指定的代码。

以下是创建 PostgreSQL 触发器的基本语法：

```
1 CREATE TRIGGER trigger_name [BEFORE|AFTER|INSTEAD OF] event_name ON
   table_name
2 FOR EACH ROW
3 EXECUTE FUNCTION function_name();
```

其中，`trigger_name` 是触发器的名称，`event_name` 是触发器要响应的事件，`table_name` 是触发器要绑定的表名，`function_name()` 是触发器要执行的函数。

以下是一些常见的 PostgreSQL 触发器事件：

- `INSERT`：在插入数据之前或之后触发。
- `UPDATE`：在更新数据之前或之后触发。
- `DELETE`：在删除数据之前或之后触发。

以下是一些常见的 PostgreSQL 触发器类型：

- `BEFORE`：在事件发生之前触发。
- `AFTER`：在事件发生之后触发。
- `INSTEAD OF`：代替事件发生。

以下是一个创建 PostgreSQL 触发器的例子：

```
1 CREATE TRIGGER example_trigger
2 AFTER INSERT ON example_table
3 FOR EACH ROW
4 EXECUTE FUNCTION example_function();
```

这个触发器会在 `example_table` 表中插入数据之后触发，并执行 `example_function()` 函数。

`INSTEAD OF` 触发器可以用于实现不可更新视图的增删改。当在一个视图上执行插入、更新或删除操作时，如果该视图不可更新，则会触发 `INSTEAD OF` 触发器，而不是执行默认的插入、更新或删除操作。

以下是一个 `INSTEAD OF` 触发器的例子，它可以用于实现不可更新视图的插入操作：

```
1 CREATE OR REPLACE FUNCTION insert_into_view() RETURNS TRIGGER AS $$
2 BEGIN
3     INSERT INTO table_name (column1, column2, ...)
4     VALUES (NEW.column1, NEW.column2, ...);
5     RETURN NEW;
6 END;
7 $$ LANGUAGE plpgsql;
8
9 CREATE TRIGGER insert_into_view_trigger
10 INSTEAD OF INSERT ON view_name
11 FOR EACH ROW
12 EXECUTE FUNCTION insert_into_view();
```

这个触发器会在插入操作发生时触发，将插入的数据插入到表中。如果视图不可更新，则会触发这个触发器，而不是执行默认的插入操作。

类似地，可以创建 `INSTEAD OF` 触发器来实现不可更新视图的更新和删除操作。这些触发器的实现方式与上面的插入触发器类似，只需要将触发器的类型从 `INSERT` 改为 `UPDATE` 或 `DELETE` 即可。

触发器可以用于实现各种功能，例如：

- 检查约束。
- 数据审计。
- 数据同步。
- 数据加密。
- 数据备份

## 四、课内实验

1. 为worker表（参照实验十一）建立触发器T1, 当插入或是更新表中数据时，保证所操作的记录的sage 值大于0。（参考代码如下：）

```
1  -- mysql不支持同时建立两个
2  DELIMITER $$
3
4  CREATE TRIGGER check_sage_trigger_insert
5  BEFORE INSERT ON worker
6  FOR EACH ROW
7  BEGIN
8      DECLARE msg VARCHAR(255);
9
10     IF NEW.Sage <= 0 THEN
11         SET msg = 'Sage must be greater than 0';
12         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
13     END IF;
14 END $$
15
16 DELIMITER ;
17
18 DELIMITER $$
```

```

19
20 CREATE TRIGGER check_sage_trigger_update
21 BEFORE UPDATE ON worker
22 FOR EACH ROW
23 BEGIN
24     DECLARE msg VARCHAR(255);
25
26     IF NEW.Sage <= 0 THEN
27         SET msg = 'Sage must be greater than 0';
28         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
29     END IF;
30 END $$
31
32 DELIMITER ;

```

执行结果如下：

查询	信息
CREATE TRIGGER check_sage_trigger BEFORE INSERT ON Worker FOR EACH ROW BEGIN DECLARE msg VARCHAR(255);	Affected rows: 0

2. 为 worker 表建立触发器T2, 禁止删除编号为00001的CEO

```

1 DELIMITER $$
2 CREATE TRIGGER forbid_delete_ceo_trigger
3 BEFORE DELETE ON worker
4 FOR EACH ROW
5 BEGIN
6     DECLARE msg VARCHAR(255);
7
8     IF OLD.Number = '00001' THEN
9         SET msg = 'CEO cannot be deleted';
10        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
11    END IF;
12 END $$
13
14 DELIMITER ;

```

执行结果如下：

查询	信息
CREATE TRIGGER forbid_delete_ceo_trigger BEFORE DELETE ON Worker FOR EACH ROW BEGIN DECLARE msg VARCHAR(255);	Affected rows: 0

3. worker 表中的人员的编号是不可改变的，创建触发器 T3 实现更新中编号的不可改变性。

```

1 DELIMITER $$
2 CREATE TRIGGER forbid_change_number_trigger
3 BEFORE UPDATE ON worker
4 FOR EACH ROW
5 BEGIN
6     DECLARE msg VARCHAR(255);
7

```

```

8      IF NEW.Number != OLD.Number THEN
9          SET msg = 'Number cannot be changed';
10         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
11     END IF;
12 END $$
13
14 DELIMITER ;

```

执行结果如下：

查询	信息
CREATE TRIGGER forbid_change_number_trigger BEFORE UPDATE ON Worker FOR EACH ROW BEGIN DECLARE msg VARCHAR(255);	Affected rows: 0

4. 演示违反 T1 触发器的约束的插入操作。

执行插入语句：

```
1 INSERT INTO worker VALUES ('00003', 'a', 'M', -1, '科技部');
```

得到以下报错：

信息
1644 - Sage must be greater than 0

5. 演示违反 T1 触发器的约束的更新操作。

执行更新语句：

```
1 UPDATE worker set Sage = Sage-15;
```

得到以下报错：

信息
1644 - Sage must be greater than 0

6. 演示违反 T2 触发器的约束的删除操作。

执行下面语句：

```
1 DELETE FROM worker WHERE Number = '00001';
```

得到以下报错：

信息
1644 - CEO cannot be deleted

7. 演示违反 T3 触发器的约束的更新操作。

执行语句：

```
1 UPDATE worker set Number = '00006';
```

得到以下报错：

信息

1644 - Number cannot be changed

## 8. 演示 INSTEAD OF 触发器在不可更新视图上的运用

Mysql没有instead of触发器。但是大致思路如下：

```
1 CREATE VIEW view1 AS
2 SELECT * FROM worker;
3
4 DELIMITER $$
5
6 CREATE TRIGGER view1_insert_trigger
7 INSTEAD OF INSERT ON view1
8 FOR EACH ROW
9 BEGIN
10     INSERT INTO worker(Number, Name, Sex, Sage, Department) VALUES
11     (NEW.Number, NEW.Name, NEW.Sex, NEW.Sage, NEW.Department);
12 END $$
13 DELIMITER ;
```

首先创建一个视图 `view1`，然后创建一个instead of触发器，检测到插入操作时将该数据插入到 `worker`表中。

## 五、自我实践

### 1. 建立一个在worker表上的触发器T4, 要求插入记录的sage值必须比表中已记录的最大 sage值大

```
1 DELIMITER $$
2 CREATE TRIGGER check_change_sage_trigger
3 BEFORE INSERT ON worker
4 FOR EACH ROW
5 BEGIN
6     DECLARE maxSage INT;
7
8     -- 获取表中已记录的最大 Sage 值
9     SELECT MAX(Sage) INTO maxSage FROM worker;
10
11     -- 如果插入记录的 Sage 小于或等于最大 Sage，则抛出异常
12     IF NEW.Sage <= maxSage THEN
13         SIGNAL SQLSTATE '45000'
14         SET MESSAGE_TEXT = 'Sage must be greater than the maximum Sage
15         in the table';
16     END IF;
17 END $$
18 DELIMITER ;
```

演示违反 T4 触发器的约束的插入操作：

```
1 | INSERT INTO worker VALUES ('00003', 'a', 'M', 10, '科技部');
```

得到报错:

信息

1644 - Sage must be greater than the maximum Sage in the table

2. 建立一个在 worker 表上的触发器 T5, 要求当更新一个记录的时候, 表中记录的 sage 值要比老记录的 sage 值大, 因为一般工资级别只能升不能降

```
1 | CDELIMITER $$
2 | CREATE TRIGGER check_update_sage_trigger
3 | BEFORE UPDATE ON worker
4 | FOR EACH ROW
5 | BEGIN
6 |     IF NEW.Sage <= OLD.Sage THEN
7 |         SIGNAL SQLSTATE '45000'
8 |         SET MESSAGE_TEXT = 'New Sage must be greater than old Sage in
the table';
9 |     END IF;
10 | END $$
11 |
12 | DELIMITER ;
```

演示违反 T5 触发器的约束的更新操作:

```
1 | UPDATE worker SET Sage = sage-2;
```

得到报错:

信息

1644 - New Sage must be greater than old Sage in the table