触发器的创建与管理

ougd . ai . wangjie

触发器是一种特殊的存储过程，其特殊之处在于触发器不是调用执行，而是当有关事件发生时被激发自动执行的。激发事件一般是对数据表的操作而引发的。

一、触发器的定义与类型

MySQL 触发器（Trigger）是一种特殊类型的存储过程，它在语句事件（INSERT、UPDATE、DELETE）执行时，触发器就会发生。与存储过程相比，触发器主要是通过事件触发从而被执行，用于处理各种复杂操作；而存储过程是通过存储过程名字被直接调用。同时注意触发器不能传递参数也不能接受参数。

二、创建触发器

在 MySQL 中，创建触发器语法如下：

```
CREATE TRIGGER trigger_name
trigger_time
trigger_event ON tbl_name
FOR EACH ROW
trigger_stmt
```

其中：

trigger_name：标识触发器名称，用户自行指定；

trigger_time：标识触发时机，取值为 BEFORE 或 AFTER；

trigger_event：标识触发事件，取值为 INSERT、UPDATE 或 DELETE；

tbl_name：标识建立触发器的表名，即在哪张表上建立触发器；

trigger_stmt：触发器程序体，可以是一句 SQL 语句，或者用 BEGIN 和 END 包含的多条语句。

由此可见，可以建立 6 种触发器，即：BEFORE INSERT、BEFORE UPDATE、BEFORE DELETE、AFTER INSERT、AFTER UPDATE、AFTER DELETE。

另外有一个限制是不能同时在一个表上建立 2 个相同类型的触发器，因此在一个表上最多建立 6 个触发器。

INSERT 型触发器：插入某一行时激活触发器，可能通过 INSERT、LOAD DATA、REPLACE 语句触发；

UPDATE 型触发器：更改某一行时激活触发器，可能通过 UPDATE 语句触发；

DELETE 型触发器：删除某一行时激活触发器，可能通过 DELETE、REPLACE 语句触发。

MySQL 除了对 INSERT、UPDATE、DELETE 基本操作进行定义外，还定义了 LOAD DATA 和 REPLACE 语句，这两种语句也能引起上述 6 中类型的触发器的触发。

LOAD DATA 语句用于将一个文件装入到一个数据表中，相当与一系列的 INSERT 操作。

REPLACE 语句一般来说和 INSERT 语句很像，只是在表中有 primary key 或 unique 索引时，如果插入的数据和原来 primary key 或 unique 索引一致时，会先删除原来的数据，然后增加一条新数据，也就是说，一条 REPLACE 语句有时候等价于一条。

INSERT 语句，有时候等价于一条 DELETE 语句加上一条 INSERT 语句。

1. 建立一个简单的 After 触发器。

例 12-15 设计一个触发器，其作用是当用户往数据库（students_courses）学生（students）表中成功插入一条记录时，更新 stu_zhy 中 stucount 字段对于专业人数的统计数据。

在开始之前，需要先建立一个表 stu_zhy，代码如下：

```
CREATE TABLE stu_zhy (
  zhy varchar(30) NOT NULL,          /*专业名称*/
  stucount int(11) DEFAULT NULL      /*专业人数*/
);
```

建立触发器的语句如下：

```
CREATE TRIGGER tri_students_insok
    AFTER INSERT
    ON students
    FOR EACH ROW
BEGIN
  DECLARE cont INT;
  DECLARE v_zhy VARCHAR(30);
  SET cont = (SELECT Count(*) FROM students WHERE zhy=NEW.zhy);
  SELECT zhy INTO v_zhy FROM stu_zhy WHERE zhy=NEW.zhy;
  IF v_zhy IS NULL
    THEN
      INSERT INTO stu_zhy VALUES(NEW.zhy,cont);
    ELSE
      UPDATE stu_zhy SET stuCount = cont WHERE zhy =NEW.zhy;
  END IF;
END;
```

上面语句在当前数据库（students_courses）中成功执行之后，就会在表 students 中建立了一个对插入（INSERT）操作建立的后触发器--tri_students_insok。

我们需要使用对表 students 执行插入（INSERT）操作才能激活触发器的执行。下面语句激活触发器。

```
INSERT students
     VALUES('201100030026','周浩天','男','通讯技术',2011,'电子工程')
```

其运行效果如图 12-11 所示，其中增加了我们定义的消息。

```
mysql> SELECT * FROM stu_zhy WHERE zhy='商务英语';
+----------+----------+
| zhy      | stucount |
+----------+----------+
| 商务英语 |        3 |
+----------+----------+
1 row in set

mysql> SELECT * FROM students WHERE zhy='商务英语';
+--------------+--------+----+----------+---------+------+
| sno          | sname  | xb | zhy      | in_year | dept |
+--------------+--------+----+----------+---------+------+
| 200900020007 | 高明明 | 女 | 商务英语 |    2009 | 外语 |
| 201000020001 | 李小可 | 女 | 商务英语 |    2010 | 外语 |
+--------------+--------+----+----------+---------+------+
2 rows in set

mysql> SELECT * FROM stu_zhy WHERE zhy='商务英语';
Empty set

mysql> INSERT students
VALUES('201100030026','周浩天','男','商务英语',2011,'电子工程');

Query OK, 1 row affected

mysql> SELECT * FROM students WHERE zhy='商务英语';
+--------------+--------+----+----------+---------+----------+
| sno          | sname  | xb | zhy      | in_year | dept     |
+--------------+--------+----+----------+---------+----------+
| 200900020007 | 高明明 | 女 | 商务英语 |    2009 | 外语     |
| 201000020001 | 李小可 | 女 | 商务英语 |    2010 | 外语     |
| 201100030026 | 周浩天 | 男 | 商务英语 |    2011 | 电子工程 |
+--------------+--------+----+----------+---------+----------+
3 rows in set

mysql> SELECT * FROM stu_zhy WHERE zhy='商务英语';
+----------+----------+
| zhy      | stucount |
+----------+----------+
| 商务英语 |        3 |
+----------+----------+
1 row in set
```

图 12-11 简单插入操作后触发实例

2．使用 BEFORE 的触发器

例 12-16 在学生（students）表上创建一个触发器，当删除一条学生数据记录时，触发器将被删除的记录在“students_copy”表中插入新数据记录作为备份。

创建表“students_copy”语句如下：

```
CREATE TABLE students (
  sno char(12) COLLATE utf8_bin NOT NULL,
  sname char(8) COLLATE utf8_bin NOT NULL,
  xb char(2) COLLATE utf8_bin DEFAULT '男',
```

```
  zhy varchar(30) COLLATE utf8_bin DEFAULT NULL,
  in_year int(11) DEFAULT NULL,
  dept varchar(30) COLLATE utf8_bin DEFAULT NULL,
  PRIMARY KEY (sno)
);
```

然后执行如下语句建立触发器。

```
CREATE TRIGGER Tri_students_InsCopy
    BEFORE DELETE
    ON students
    FOR EACH ROW
BEGIN
   /*首先声明六个变量*/
  DECLARE var_sno char(12);
  DECLARE var_sname char(8);
  DECLARE var_xb char(2);
  DECLARE var_zhy varchar(30);
  DECLARE var_inyear int;
  DECLARE var_dept varchar(30);
   /*将保存在 Inserted 表中的记录内容保存在对应的变量中*/
  SELECT sno,sname,xb,zhy,in_year,dept
    INTO var_sno,var_sname,var_xb,var_zhy,var_inyear,var_dept
  FROM students WHERE sno= OLD.sno;
  /*往 students_copy 表中添加插入的新记录。*/
  INSERT INTO students_copy(sno,sname,xb,zhy,in_year,dept)
    VALUES(var_sno,var_sname,var_xb,var_zhy,var_inyear,var_dept);
END;
```

使用下面语句进行测试：

```
    DELETE FROM students WHERE sno='201100030026';
```

```
mysql> SELECT * FROM students WHERE zhy='商务英语';
+--------------+--------+----+----------+---------+------+
| sno          | sname  | xb | zhy      | in_year | dept |
+--------------+--------+----+----------+---------+------+
| 200900020007 | 高明明 | 女 | 商务英语 |    2009 | 外语 |
| 201000020001 | 李小可 | 女 | 商务英语 |    2010 | 外语 |
| 201100030026 | 周浩天 | 男 | 商务英语 |    2011 | 外语 |
+--------------+--------+----+----------+---------+------+
3 rows in set

mysql> SELECT * FROM students_copy WHERE zhy='商务英语';
Empty set

mysql>  DELETE FROM students WHERE sno='201100030026';
Query OK, 1 row affected

mysql> SELECT * FROM students WHERE zhy='商务英语';
+--------------+--------+----+----------+---------+------+
| sno          | sname  | xb | zhy      | in_year | dept |
+--------------+--------+----+----------+---------+------+
| 200900020007 | 高明明 | 女 | 商务英语 |    2009 | 外语 |
| 201000020001 | 李小可 | 女 | 商务英语 |    2010 | 外语 |
+--------------+--------+----+----------+---------+------+
2 rows in set

mysql> SELECT * FROM students_copy WHERE zhy='商务英语';
+--------------+--------+----+----------+---------+------+
| sno          | sname  | xb | zhy      | in_year | dept |
+--------------+--------+----+----------+---------+------+
| 201100030026 | 周浩天 | 男 | 商务英语 |    2011 | 外语 |
+--------------+--------+----+----------+---------+------+
1 row in set
```

三、NEW 与 OLD 的使用

上述示例中使用了 NEW 关键字，和 MS SQL Server 中的 INSERTED 和 DELETED 类似，MySQL 中定义了 NEW 和 OLD，用来表示触发器的所在表中，触发了触发器的那一行数据。具体地：

1. 在 INSERT 型触发器中，NEW 用来表示将要（BEFORE）或已经（AFTER）插入的新数据。
2. 在 UPDATE 型触发器中，OLD 用来表示将要或已经被修改的原数据，NEW 用来表示将要或已经修改为的新数据。
3. 在 DELETE 型触发器中，OLD 用来表示将要或已经被删除的原数据。

使用方法：

```
NEW.columnName
```

columnName 为相应数据表某一列名。另外，OLD 是只读的，而 NEW 则可以在触发器中使用 SET 赋值，这样不会再次触发触发器，造成循环调用（如每插入一个学生前，都在其学号前加“2013”）。

四、管理 DML 触发器

1. 查看触发器

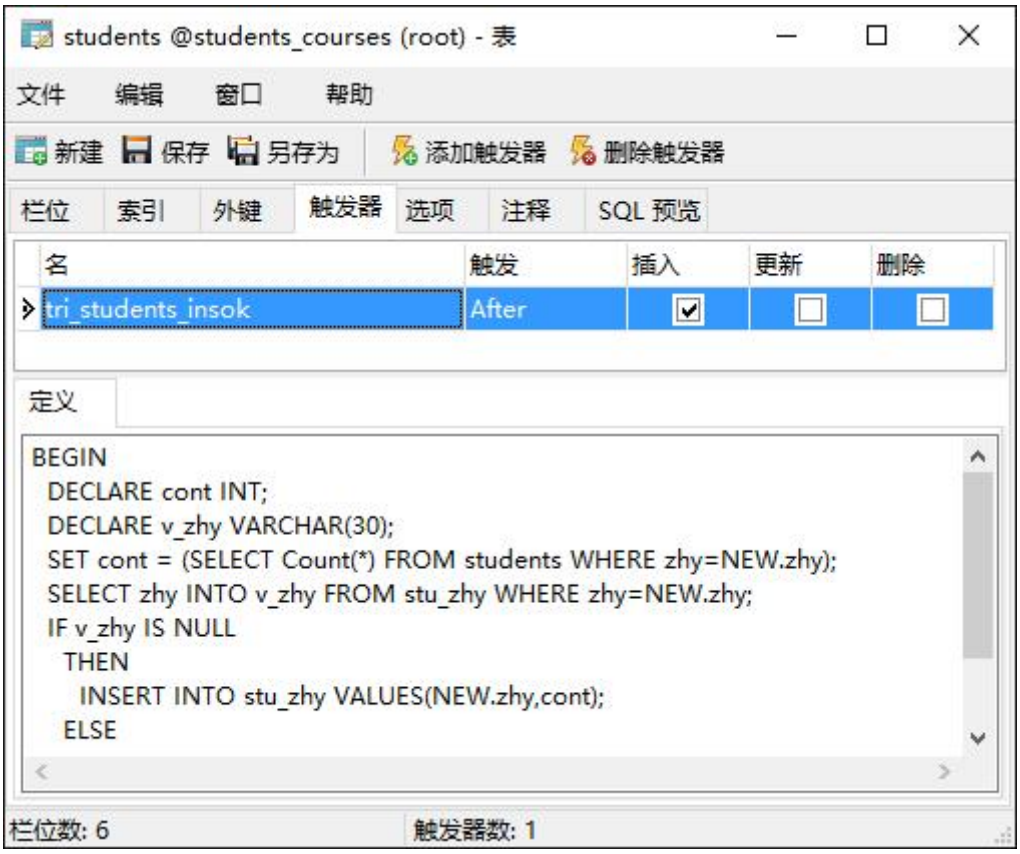
和查看数据库（show databases;）查看表格（show tables;）一样，查看触发器的语法如下：

```
SHOW TRIGGERS [FROM schema_name];
```

其中，schema_name 即 Schema 的名称，在 MySQL 中 Schema 和 Database 是一样的，也就是说，可以指定数据库名，这样就不必先“USE database_name;”了。

(2) 使用 Navicat 查看触发器信息

使用 Navicat 查看触发器信息其操作步骤如下：运行 Navicat 打开到服务器连接，双击“students_courses” →右击 “students”表 →“设计表”代开表设计对话框。



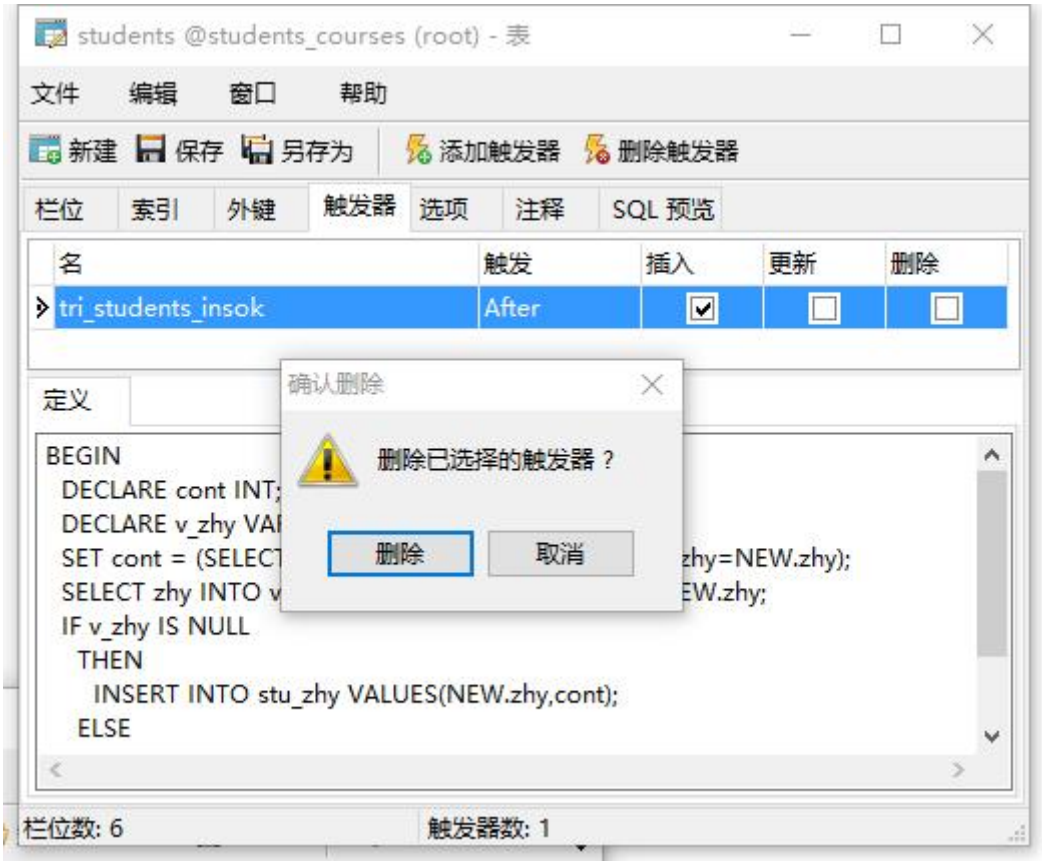
2. 删除触发器

当用户不再需要触发器后就可以将其删除，只有触发器所有者才有权利删除触发器。删除触发器所在的表时，MySQL 数据库系统将自动删除与该表相关的触发器。删除已创建的触发器有两种方法：

(1) 使用 DROP TRIGGER 来删除触发器，和删除数据库、删除表格一样，删除触发器的语法如下：

```
DROP TRIGGER [IF EXISTS] [schema_name.]trigger_name
```

(2) 使用 Navicat 删除触发器，右击要删除的触发器所在表→“设计表”代开表设计对话框，在出的对话框中选择“触发器”，选中要删除的触发器，点击“删除触发器”，再在弹出的对话框中点击“删除”就可以将触发器删除。如图



3. 修改触发器

修改触发器包括修改触发器的名称和修改触发器的内容，可以使用系统存储过程修改触发器的名称，还可以使用 Management Studio 和 Transact-SQL 修改触发器的内容。

使用 Navicat 修改触发器的方法：

运行 Navicat 打开到服务器连接，双击“students_courses” →右击 “students”表 →“设计表”代开表设计对话框，在对话框中修改属性和代码，之后点击“保存”即可。

MySQL 中没有直接修改触发器的命令，可以将原触发器删除然后重新创建。

五、触发器使用注意

在创建触发器以前必须仔细考虑到以下几个方面：

(1) 一个触发器只能对应一个表，这是由触发器的机制决定的。

(2) 触发器是数据库或表中的对象，所以其命名必须符合 MySQL 命名规则。

(3) CREATE TRIGGER 语句必须是批处理的第一个语句。

(4) 表的所有者具有创建触发器的缺省权限，表的所有者不能把该权限传给其它用户。

(5) 尽管在触发器的 T-SQL 语句中可以参照其它数据库中的对象，但是触发器只能创建在当前数据库中。

(6) 虽然触发器可以参照视图或临时表，但不能在视图或临时表上创建触发器，而只能在基表或在创建视图的表上创建触发器。

(7) 当创建一个触发器时必须指定触发器的名字在哪一个表上，定义激活触发器的修改语句(如 INSERT、DELETE、UPDATE) ，当然两个或三个不同的修改语句也可以都触发同一个触发器

(8) 数据库和表中尽量少用触发器多用存储过程，以免出现不可预知的错误。