
8.4.1 触发器介绍

MySQL 是使用 CREATE TRIGGER 语句创建触发器的。语法格式如下：

```
CREATE <触发器名> < BEFORE | AFTER >  
<INSERT | UPDATE | DELETE >  
ON <表名> FOR EACH ROW<触发器主体>
```

语法说明如下。

1. 触发器名

触发器的名称，触发器在当前数据库中必须具有唯一的名称。如果要在某个特定数据库中创建，名称前面应该加上数据库的名称。

2. INSERT | UPDATE | DELETE

触发事件，用于指定激活触发器的语句的种类。

注意：三种触发器的执行时间如下。

INSERT：将新行插入表时激活触发器。例如，INSERT 的 BEFORE 触发器不仅能被 MySQL 的 INSERT 语句激活，也能被 LOAD DATA 语句激活。

DELETE：从表中删除某一行数据时激活触发器，例如 DELETE 和 REPLACE 语句。

UPDATE：更改表中某一行数据时激活触发器，例如 UPDATE 语句。

3. BEFORE | AFTER

BEFORE 和 AFTER，触发器被触发的时刻，表示触发器是在激活它的语句之前或之后触发。若希望验证新数据是否满足条件，则使用 BEFORE 选项；若希望在激活触发器的语句执行之后完成几个或更多的改变，则通常使用 AFTER 选项。

4. 表名

与触发器相关联的表名，此表必须是永久性表，不能将触发器与临时表或视图关联起来。在该表上触发事件发生时才会激活触发器。同一个表不能拥有两个具有相同触发时刻和事件的触发器。例如，对于一张数据表，不能同时有两个 BEFORE UPDATE 触发器，但可以有一个 BEFORE UPDATE 触发器和一个 BEFORE INSERT 触发器，或一个 BEFORE UPDATE 触发器和一个 AFTER UPDATE 触发器。

5. 触发器主体

触发器动作主体，包含触发器激活时将要执行的 MySQL 语句。如果要执行多个语句，需要使用 BEGIN...END 复合语句结构。

6. FOR EACH ROW

一般是指行级触发，对于受触发事件影响的每一行都要激活触发器的动作。例如，使用 INSERT 语句向某个表中插入多行数据时，触发器会对每一行数据的插入都执行相应的触发器动作。

上述创建语法可见，触发器创建需要四个要素

1. 监视地点 (table)
2. 监视事件 (insert/update/delete)
3. 触发时间 (after/before)

BEFORE 关键字可使触发器在涉及的数据修改之前执行。可使用 BEFORE 触发器捕获无效数据条目并在写入表之前对其进行更正或拒绝。

- BEFORE INSERT：在添加新数据之前触发
- BEFORE UPDATE：在使用新数据更新（或覆盖）现有数据之前触发
- BEFORE DELETE：在删除数据之前触发

AFTER 关键字定义在数据修改成功后执行的触发器。可使用 AFTER 触发器记录或审计数据库中的数据修改。

- AFTER INSERT : 在添加新数据之后触发
- AFTER UPDATE : 在使用新数据更新（或覆盖）现有数据之后触发
- AFTER DELETE : 在删除数据之后触发

4. 触发事件 (insert/update/delete)

注意：每个表都支持 INSERT、UPDATE 和 DELETE 的 BEFORE 与 AFTER，因此每个表最多支持 6 个触发器。每个表的每个事件每次只允许有一个触发器。单一触发器不能与多个事件或多个表关联。

在触发事件中可以引用行变量

- 在监视事件上执行 insert 操作后会有一个新行，如果在触发事件中需要用到这个新行的变量，可以用 new 关键字表示。

- 在监视事件上执行 delete 操作后会有一个旧行，如果在触发事件中需要用到这个旧行的变量，可以用 old 关键字表示。

- 在监视事件上执行 update 操作后原纪录是旧行，新纪录是新行，可以使用 new 和 old 关键字来分别操作。