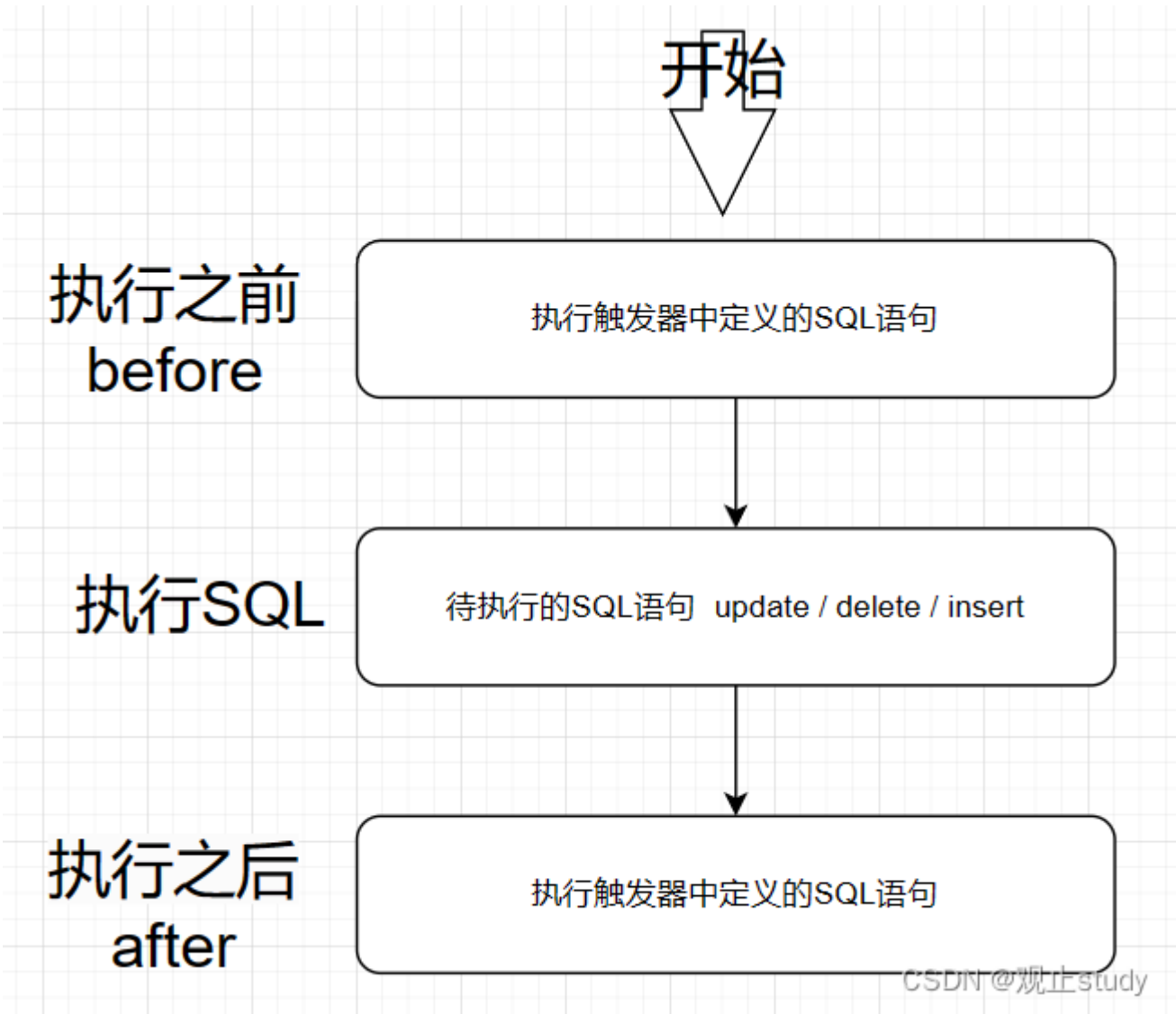


# 【MySQL】触发器 (十二)

🚗 MySQL学习·第十二站~  
📌 本文已收录至专栏：[MySQL通关路](#)  
❤️ 文末附全文思维导图，感谢各位点赞收藏支持~  
★ 学习汇总贴，超详细思维导图：[【MySQL】学习汇总\(完整思维导图\)](#)

## 一.引入

触发器是与表有关的数据库对象，作用在 `insert/update/delete` 语句执行之前( `BEFORE` )或之后( `AFTER` )，**自动触发并执行触发器中定义的SQL语句集合**。它可以协助应用在数据库端确保数据的完整性，日志记录，数据校验等操作。



可以在触发器内定义的SQL语句中使用 `OLD`（原始记录）或 `NEW`（新增记录）来引用所操作行的数据。目前触发器只支持行级触发，不支持语句级触发，即每影响一行数据便会触发一次。

触发器类型	NEW 和 OLD含义
<code>INSERT</code> 型触发器	<code>NEW</code> 表示将要或者已经新增的数据
<code>UPDATE</code> 型触发器	<code>OLD</code> 表示修改之前的数据， <code>NEW</code> 表示将要或已经修改后的数据
<code>DELETE</code> 型触发器	<code>OLD</code> 表示将要或者已经删除的数据

id	name
1	javaEE
2	Spring
3	MySQL
4	Cloud

执行 update user set name = 'SpringBoot' where id = 2

	id	name
old	old.id = 2	old.name=Spring
new	new.id = 2	new.name = SpringBoot

字段名

变化的数据行

变化数据行中的字段值  
新旧行.字段名

CSDN @观止study

## 二.相关语法

### (1) 创建

- 语法

```
CREATE TRIGGER 触发器名称
BEFORE/AFTER INSERT/UPDATE/DELETE
ON 表名 FOR EACH ROW
BEGIN
-- SQL语句 ;
END;
```

```
1 -- 创建插入之前校验触发器
2 CREATE TRIGGER test_age
3 BEFORE INSERT
4 ON user FOR EACH ROW
5 BEGIN
6     -- 插入后立即查询当前表所有数据
7     if new.age > 120 THEN
8         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = '非法年龄';
9     END IF;
10 END;
11
12 insert into user (name,age,gender) values ('观',125,'1')
```

数据校验，抛出自定义异常

信息 概况 状态

[SQL]insert into user (name,age,gender) values('观',125,'1')

[Err] 1644 - 非法年龄

CSDN @观止study

#### 注意事项

- 触发器中的SQL语句不能出现 `SELECT * FROM TABLE` 形式的查询，因为会返回一个结果集，使用时会抛出错误 `Not allowed to return a result set from a trigger`，可以使用 `SELECT INTO` 为变量设置值。
- 在使用插入/更新触发器时，由于MySQL的写锁，无法在触发器内再次定义对当前表的更新或插入SQL语句。

## (2) 查看

- 语法

```
SHOW TRIGGERS;
```

```
1 SHOW TRIGGERS;
```

信息	结果1	概况	状态
Trigger	Event	Table	Statement
test_age	INSERT	user	BEGIN-- 插入后立即查询当前表所有数据if new.age > 120 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = '非法年龄';END IF;END BEFORE

## (3) 删除

- 语法

```
DROP TRIGGER [数据库名称.]触发器名称 ;
```

如果没有指定数据库名称，默认为当前数据库。

```
2 -- 删除触发器
3 DROP TRIGGER study.test_age ;
4 -- 查看触发器
5 SHOW TRIGGERS;
```

信息	结果1	概况	状态
Trigger	Event	Table	Statement
(Null)	(Null)	(Null)	(Null)

已删除

## 三.日志记录示例

使用触发器可以快捷的记录表数据变更日志。接下来我们通过触发器记录 user 表的数据变更日志，将变更日志插入到日志表 user\_logs中, 包含增加, 修改, 删除, 学习三种触发器的使用。

### (1) 创建表

1. 创建 user 表以及 user\_logs表

```
-- 用户表 user
CREATE TABLE user (
  id int NOT NULL AUTO_INCREMENT COMMENT '主键',
  name varchar(50) NOT NULL COMMENT '用户名',
  phone varchar(11) NULL COMMENT '手机号',
  email varchar(100) DEFAULT NULL COMMENT '邮箱',
  profession varchar(11) DEFAULT NULL COMMENT '专业',
  age tinyint unsigned DEFAULT NULL COMMENT '年龄',
  gender char(1) DEFAULT NULL COMMENT '性别 , 1: 男, 2: 女',
  status char(1) DEFAULT NULL COMMENT '状态',
  createtime datetime DEFAULT NULL COMMENT '创建时间',
  PRIMARY KEY (`id`),
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='用户表';
```

```
-- 日志表 user_logs
create table user_logs(
id int(11) not null auto_increment,
operation varchar(20) not null comment '操作类型, insert/update/delete',
operate_time datetime not null comment '操作时间',
operate_id int(11) not null comment '操作的ID',
operate_params varchar(500) comment '操作参数',
primary key(`id`))
)engine=innodb default charset=utf8 COMMENT='用户日志表';
```

## (2) insert型触发器

### 2. 创建插入 insert 型触发器

```
create trigger user_insert_trigger
-- 每次在user表执行完插入操作之后触发
after insert on user for each row
begin
-- 在日志表插入操作日志
-- 通过 new 可以获得新插入的数据行记录
insert into user_logs(id, operation, operate_time, operate_id, operate_params)
VALUES
(null, 'insert', now(), new.id, concat('插入的数据内容为: id=',new.id,',name=',new.name, ',
phone=',NEW.phone, ', email=', NEW.email, ', profession=', NEW.profession));
end;
```

测试:

```
-- 插入一条数据到user表
insert into user(id, name, phone, email, profession, age, gender, status,
createtime) VALUES (26,'三皇子','18809091212','erhuangzi@163.com','软件工
程',23,'1','1',now());
```

```
12
13 insert into user(id, name, phone, email, profession, age, gender, status,
14 createtime) VALUES (26,'三皇子','18809091212','erhuangzi@163.com','软件工
15 程',23,'1','1',now());
```

id	operation	operate_time	operate_id	operate_params
1	insert	2023-08-01 09:15:25	26	插入的数据内容为: id=26,name=三皇子, phone=18809091212, email=erhuangzi@163.com, profession=软件工程

插入成功的同时自动往user\_logs表中新增了一条记录

信息 概况 状态

[SQL]insert into user(id, name, phone, email, profession, age, gender, status, createtime) VALUES (26,'三皇子','18809091212','erhuangzi@163.com','软件工程',23,'1','1',now());

受影响的行: 1  
时间: 0.003s

CSDN @观止study

## (3) update型触发器

### 3. 创建修改 update 型触发器

```

create trigger user_update_trigger
-- 每次在user表执行完更新操作之后触发
after update on user for each row
begin
insert into user_logs(id, operation, operate_time, operate_id, operate_params)
VALUES
-- 在日志表插入操作日志
-- 通过 new 可以获得修改之后的数据行记录
-- 通过 old 可以获得修改之前的数据行记录
(null, 'update', now(), new.id, concat('更新之前的数据: id=', old.id, ', name=', old.name, ', phone=',
old.phone, ', email=', old.email, ', profession=', old.profession, ' | 更新之后的数据:
id=', new.id, ', name=', new.name, ', phone=', NEW.phone, ', email=', NEW.email, ', profession=',
NEW.profession));
end;

```

测试

```

-- 对user表执行更新操作
update user set name = '品如' where id = 26;

```

```

16 update user set name = '品如' where id = 26;

```

id	operation	operate_time	operate_id	operate_params
1	insert	2023-08-01 09:15:25	26	插入的数据内容为: id=26,name=三皇子, phone=18809091212, email=erhuangzi@163.com, profession=软件工程
2	update	2023-08-01 09:22:04	26	更新之前的数据: id=26,name=三皇子, phone=18809091212, email=erhuangzi@163.com, profession=软件工程   更新之后的数据: id=26,name=品如,

信息 概况 状态

[SQL]update user set name = '品如' where id = 26;  
 受影响的行: 1  
 时间: 0.008s

执行更新操作成功，日志表成功记录日志

CSDN @观止study

## (4) delete型触发器

### 4. 创建删除 delete 型触发器

```

create trigger user_delete_trigger
-- 每次在user表执行完删除操作之后触发
after delete on user for each row
begin
insert into user_logs(id, operation, operate_time, operate_id, operate_params)
VALUES
-- 在日志表插入操作日志
-- 通过 old 可以获得已经删除的数据行记录
(null, 'delete', now(), old.id, concat('删除之前的数据: id=', old.id, ', name=', old.name, ', phone=',
old.phone, ', email=', old.email, ', profession=', old.profession));
end;

```

测试:

```

delete from user where name = '品如';

```

13 delete from user where name = '品如';|

id	operation	operate_time	operate_params
1	insert	2023-08-01 09:15:25	26 插入的数据内容为: id=26,name=三皇子, phone=18809091212, email=erhuangzi@163.com, profession=软件工
2	update	2023-08-01 09:22:04	26 更新之前的数据: id=26,name=三皇子, phone=18809091212, email=erhuangzi@163.com, profession=软件工
3	delete	2023-08-01 09:26:45	26 删除之前的数据: id=26,name=品如, phone=18809091212, email=erhuangzi@163.com, profession=软件工

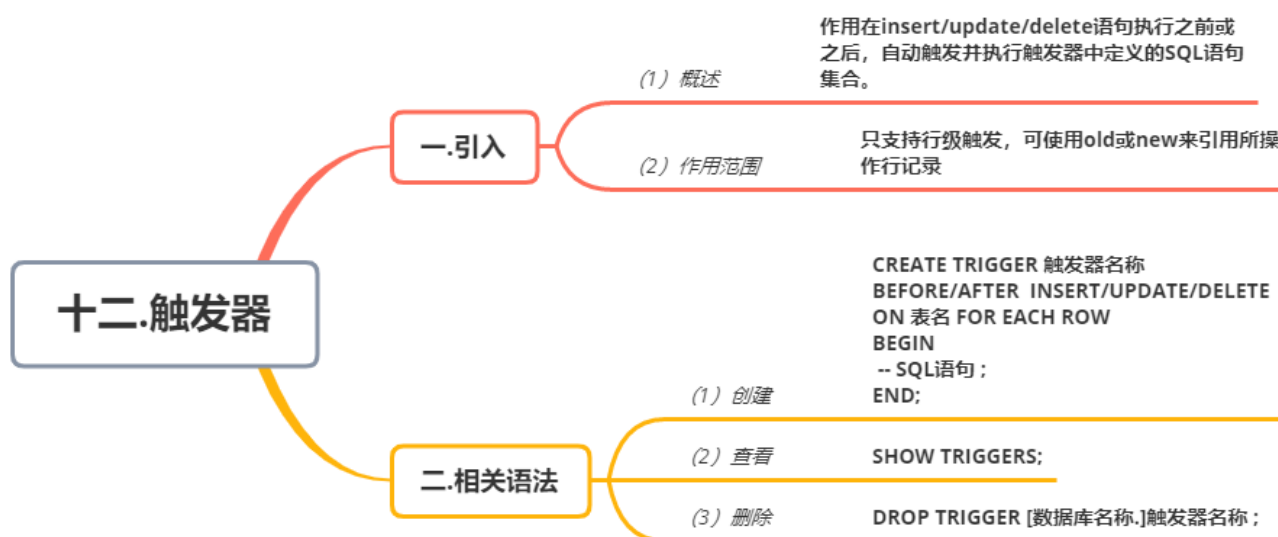
信息 概况 状态

删除成功，并且成功记录日志

[SQL]delete from user where name = '品如';  
受影响的行: 1  
时间: 0.007s

CSDN @观止study

## 四.全文概览



CSDN @观止study