



5.6 触发器

☞ 触发器(Trigger)

用户定义在关系表上的一类由事件驱动的特殊存储过程。

- * 用户对表的操作会自动激活相应的触发器，此过程由DBMS控制；
- * 类似于约束，但是比约束功能强大，可以实施比FOREIGN KEY, CHECK复杂的检查和操作；

触发器的定义：

```
CREATE TRIGGER <触发器名>  
    {BEFORE | AFTER} <触发事件> ON <表名>  
    FOR EACH {ROW | STATEMENT}  
    [WHEN <触发条件>]  
    <触发动作体>;
```



5.6 触发器

```
CREATE TRIGGER <触发器名>
  {BEFORE | AFTER} <触发事件> ON <表名>
  FOR EACH {ROW | STATEMENT}
  [WHEN <触发条件>]
  <触发动作体>
```

注:

- * 触发器可以在触发事件之前或之后执行;
- * 触发事件为INSERT/DELETE/UPDATE;
- * 按照触发动作分为**行级触发器**和**语句级触发器**;
- * 当触发器被激活时, 触发条件为真才执行触发动作体;
- * 同一个表上的多个触发器遵循如下的执行顺序: 执行该表上的BEFORE触发器; 激活触发器的SQL语句; 执行该表上的AFTER触发器。
- * 触发器不属于SQL规范, **不同的DBMS语法不同**;



5.6 触发器

[例10] 在建立Student和SC时，没有使用外码参照完整性来建立两者的联系，可使用触发器来实现数据的参照完整性：当从Student中删除学生时，SC中的相关成绩也删除掉。

以SQL Server2000为例：

```
CREATE TRIGGER del_Student ON Student
FOR DELETE AS
    DELETE FROM SC          WHERE Sno IN
    WHERE Sno NOT IN      ( SELECT Sno FROM DELETED );
    ( SELECT Sno FROM Student );
```

del_Student保证当从Student中删除数据时，SC与其保持一致。

```
DELETE FROM Student WHERE Sno='95001';
```

激活del_Student触发器，将95001的成绩也删除掉。



第五章 数据库完整性

👉 作业

P164 题6



第五章 数据库完整性

👉 本章小结

* 数据库完整性

- 实体完整性: **PRIMARY KEY**
- 参照完整性: **FOREIGN KEY**拒绝/级联删除/设置为空值
- 用户定义完整性: **NOT NULL/UNIQUE/CHECK**
- 完整性约束命名: **CONSTRAINT**
- 域: **DOMAIN**
- 触发器