数据库原理及应用

Principle and Application of Database

华东交通大学
喻佳



5.3 触发器

触发器

1、定义触发器

2、激活触发器

3、删除触发器

触发器的定义

- 触发器（Trigger）是用户定义在关系表上的一类由事件驱动的特殊过程
  - 由服务器自动激活
  - 可以进行更为复杂的检查和操作，具有更精细和更强大的数据控制能力

创建 触发器

- CREATE TRIGGER语法格式

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>

创建 触发器

事件-条件-动作

- CREATE TRIGGER语法格式

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>

创建 触发器

事件-条件-动作

- CREATE TRIGGER语法格式

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>

创建 触发器

事件-条件-动作

- CREATE TRIGGER语法格式

CREATE TRIGGER <

指明触发器激活的时间

{**BEFORE | AFTER**} <触发事件> ON <表名>

FOR EACH {ROW | STATEMENT}

［WHEN <触发条件>］ <触发动作体>

创建 触发器

事件-条件-动作

- CR 格式

  CR <触发器名>

  {BEFORE | AFTER} <触发事件> ON **<表名>**

  FOR EACH {ROW | STAT

  ［WHEN <触发条件>］ <触发

触发器只能定义在基本**表**上，而不能定义在视图上

表的拥有者方能创建触发器

创建 触发器

事件-条件-动作

- CREATE TRIGGER语法格式

CREATE　TRIGGER　<触发

{BEFORE | AFTER}　<触发事

FOR　EACH　{ROW | STATEMENT}

［WHEN <触发条件>］　<触发动作体>

触发器类型：
行级/语句触发

创建 触发器

事件-条件-动作

- CREATE TRIGGER语法格式

CREATE　TRIGGER　<触发器名>

{B　　件> ON <表名>

F　　ENT}

只有满足条件，触发体才执行

［**WHEN <触发条件>**］ **<触发动作体>**

［例］ 定义一个BEFORE行级触发器，为教师表Teacher定义完整性规则“教授的工资不得低于8000元，如果低于8000元，自动改为8000元”。

CREATE TRIGGER

BEFORE ON

/*触发事件是插入或更新操作*/

FOR EACH /*行级触发器*/

/*定义触发动作体，是PL/SQL过程块*/



CREATE TRIGGER Insert_Or_Update_Sal

BEFORE ON

/*触发事件是插入或更新操作*/

FOR /*行级触发器*/

/*定义触发动作体，是PL/SQL过程块*/



```
CREATE TRIGGER    Insert_Or_Update_Sal
    BEFORE    INSERT   OR   UPDATE   ON Teacher
                                    /*触发事件是插入或更新操作*/
    FOR                                    /*行级触发器*/
                                /*定义触发动作体，是PL/SQL过程块*/
```

```
CREATE TRIGGER    Insert_Or_Update_Sal
    BEFORE    INSERT   OR   UPDATE   ON Teacher
                                    /*触发事件是插入或更新操作*/
    FOR     EACH    ROW              /*行级触发器*/
                                /*定义触发动作体，是PL/SQL过程块*/
```

```
CREATE TRIGGER    Insert_Or_Update_Sal
    BEFORE    INSERT   OR   UPDATE   ON Teacher
                                    /*触发事件是插入或更新操作*/
    FOR     EACH    ROW              /*行级触发器*/
   AS BEGIN        /*定义触发动作体，是PL/SQL过程块*/
        IF (new.Job='教授') AND (new.Sal < 8000)  THEN
        new.Sal :=8000;
        END IF;
   END;
```

［例］定义AFTER行级触发器，当教师表Teacher的工资发生变化后就自动在工资变化表Sal_log中增加一条相应记录

［例］定义AFTER行级触发器，当教师表Teacher的工资发生变化后就自动在工资变化表Sal_log中增加一条相应记录

首先建立工资变化表Sal_log

```
CREATE TABLE Sal_log
   (  Eno    NUMERIC(4)  references teacher(eno)，
      Sal     NUMERIC(7，2)，
      Username  char(10)，
      Date   TIMESTAMP   )；
```

［例］定义AFTER行级触发器，当教师表Teacher的工资发生变化后就自动在工资变化表Sal_log中增加一条相应记录

```
CREATE TRIGGER Insert_Sal
  AFTER INSERT ON Teacher        /*触发事件是INSERT*/
  FOR EACH ROW
  AS BEGIN
    INSERT INTO Sal_log VALUES(newEno, newSal
  CURRENT_USER, CURRENT_TIMESTAMP)
  END;
```

［例］定义AFTER行级触发器，当教师表Teacher的工资发生变化后就自动在工资变化表Sal_log中增加一条相应记录

```
CREATE TRIGGER Insert_Sal
  AFTER  INSERT  ON  Teacher      /*触发事件是INSERT*/
  FOR  EACH ROW
  AS BEGIN
    INSERT INTO Sal_log VALUES( newEno, newSal,
  CURRENT_USER, CURRENT_TIMESTAMP)
  END;
```

［例］定义AFTER行级触发器，当教师表Teacher的工资发生变化后就自动在工资变化表Sal_log中增加一条相应记录

```
CREATE TRIGGER Insert_Sal
    AFTER INSERT ON Teacher      /*触发事件是INSERT*/
    FOR
```

［例］定义AFTER行级触发器，当教师表Teacher的工资发生变化后就自动在工资变化表Sal_log中增加一条相应记录

```
CREATE TRIGGER Insert_Sal
    AFTER INSERT ON Teacher      /*触发事件是INSERT*/
   FOR EACH ROW
   AS BEGIN



   END;
```

［例］定义AFTER行级触发器，当教师表Teacher的工资发生变化后就自动在工资变化表Sal_log中增加一条相应记录

```
CREATE TRIGGER Insert_Sal
    AFTER INSERT ON Teacher      /*触发事件是INSERT*/
   FOR EACH ROW
   AS BEGIN
       INSERT INTO Sal_log VALUES( new.Eno，new.Sal，
CURRENT_USER，CURRENT_TIMESTAMP);
   END;
```

```
CREATE TRIGGER Update_Sal
  AFTER UPDATE ON Teacher      /*触发事件是UPDATE */
  FOR EACH ROW
  AS BEGIN
    IF (new.Sal <> old.Sal) THEN INSERT INTO Sal_log
VALUES( new.Eno，new.Sal，CURRENT_USER，
CURRENT_TIMESTAMP);
    END IF;
END;
```

激活触发器

- 触发器的执行，是由触发事件激活的，并由数据库服务器自动执行
- 一个数据表上可能定义了多个触发器
  - 同一个表上的多个触发器激活时遵循如下的执行顺序：
    - （1） 执行该表上的BEFORE触发器；
    - （2） 激活触发器的SQL语句；
    - （3） 执行该表上的AFTER触发器。

激活触发器

［例］执行修改某个教师工资的SQL语句，激活上述定义的触发器。

```
UPDATE Teacher
SET Sal=800
WHERE Ename='陈平';
```

执行顺序是：

- 执行触发器Insert_Or_Update_Sal
- 执行SQL语句“UPDATE Teacher SET Sal=800 WHERE Ename='陈平';”
- 执行触发器Insert_Sal；
- 执行触发器Update_Sal

删除触发器

- 删除触发器的SQL语法：

  DROP TRIGGER <触发器名> ON <表名>;

- 触发器必须是一个已经创建的触发器，并且只能由具有相应权限的用户删除。

［例］ 删除教师表Teacher上的触发器Insert_Sal

DROP TRIGGER Insert_Sal ON Teacher;

触发器小结

1、触发器的定义

2、如何创建及激活触发器

3、删除触发器

谢　　谢

THANK YOU FOR YOUR WATCHING

