



南京大学软件学院
NANJING UNIVERSITY · SOFTWARE INSTITUTE



数据管理基础

ch30 断言

Software Institute
Nanjing University
Bei Jia

- SQL中，可以使用CREATE ASSERTION语句，通过声明性断言来指定更具一般性的约束。
- 可以定义涉及多个表的或聚集操作的比较复杂的完整性约束。
- 断言创建以后，任何对断言中所涉及的关系的操作都会触发关系数据库管理系统对断言的检查，任何使断言不为真值的操作都会被拒绝执行
- 如果断言很复杂，则系统在检测和维护断言的开销较高，这是在使用断言时应该注意的



- 创建断言的语句格式

- CREATE ASSERTION<断言名><CHECK 子句>
- 每个断言都被赋予一个名字，<CHECK 子句>中的约束条件与WHERE子句的条件表达式类似。

- 删除断言的语句格式为

- DROP ASSERTION <断言名>;

创建断言 1

- [例5.18] 限制数据库课程最多60名学生选修

```
CREATE ASSERTION ASSE_SC_DB_NUM
```

```
    CHECK (60 >= (select count(*))
```

/*此断言的谓词涉及聚集操作count的SQL语句*/

```
From Course,SC
```

```
Where SC.Cno=Course.Cno and
```

```
        Course.Cname ='数据库')
```

```
);
```

创建断言 2

- [例5.19]限制每一门课程最多60名学生选修

```
CREATE ASSERTION ASSE_SC_CNUM1  
CHECK(60 >= ALL (SELECT count(*)  
                  FROM SC  
                  GROUP by cno)  
);
```

/*此断言的谓词，涉及聚集操作count 和分组函数
group by的SQL语句*/



创建断言 3

- [例5.20]限制每个学期每一门课程最多60名学生选修。首先需要修改SC表的模式，增加一个“学期 (TERM) ”属性

```
ALTER TABLE SC ADD TERM DATE;
```

- 然后，定义断言：

```
CREATE ASSERTION ASSE_SC_CNUM2  
CHECK(60 >= ALL (SELECT count(*)  
                  FROM SC  
                 GROUP by cno,TERM)  
);
```





南京大学软件学院
NANJING UNIVERSITY · SOFTWARE INSTITUTE



数据管理基础

ch31 触发器

Software Institute
Nanjing University
Bei Jia

触发器

- 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
 - 触发器保存在数据库服务器中
 - 任何用户对表的增、删、改操作均由服务器自动激活相应的触发器
 - 触发器可以实施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
- 不同的RDBMS产品触发器语法各不相同



定义触发器 1

- CREATE TRIGGER语法格式

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

- 触发器又叫做事件-条件-动作 (event-condition-action) 规则。
 - 当特定的系统事件发生时，对规则的条件进行检查，如果条件成立则执行规则中的动作，否则不执行该动作。规则中的动作体可以很复杂，通常是一段SQL存储过程。



定义触发器 2

- 表的拥有者才可以在表上创建触发器
- 触发器名
 - 触发器名可以包含模式名，也可以不包含模式名
 - 同一模式下，触发器名必须是唯一的
 - 触发器名和表名必须在同一模式下
- 表名
 - 触发器只能定义在基本表上，不能定义在视图上
 - 当基本表的数据发生变化时，将激活定义在该表上相应触发事件的触发器
- 触发事件
 - 触发事件可以是INSERT、DELETE或UPDATE，也可以是这几个事件的组合
 - 还可以UPDATE OF<触发列，...>，即进一步指明修改哪些列时激活触发器
 - AFTER/BEFORE是触发的时机
 - ◆ AFTER表示在触发事件的操作执行之后激活触发器
 - ◆ BEFORE表示在触发事件的操作执行之前激活触发器
- 触发器类型
 - 行级触发器 (FOR EACH ROW)
 - 语句级触发器 (FOR EACH STATEMENT)
- 触发条件
 - 触发器被激活时，只有当触发条件为真时触发动作体才执行；否则触发动作体不执行。
 - 如果省略WHEN触发条件，则触发动作体在触发器激活后立即执行
- 触发动作体
 - 触发动作体可以是一个匿名PL/SQL过程块，也可以是对已创建存储过程的调用
 - 如果是行级触发器，用户可以在过程体中使用NEW和OLD引用事件之后的新值和事件之前的旧值
 - 如果是语句级触发器，则不能在触发动作体中使用NEW或OLD进行引用
 - 如果触发动作体执行失败，激活触发器的事件就会终止执行，触发器的目标表或触发器可能影响的其他对象不发生任何变化



触发器类型

- 例如,在例5.11的TEACHER表上创建一个AFTER UPDATE触发器, 触发事件是UPDATE语句:
 - UPDATE TEACHER SET Deptno=5;
 - 假设表TEACHER有1000行
 - ◆ 如果是语句级触发器, 那么执行完该语句后, 触发动作只发生一次
 - ◆ 如果是行级触发器, 触发动作将执行1000次



触发器示例 1

- [例5.21]当对表SC的Grade属性进行修改时，若分数增加了10%则将此次操作记录到下面表中： SC_U (Sno,Cno,Oldgrade,Newgrade) 其中Oldgrade是修改前的分数， Newgrade是修改后的分数。

```
CREATE TRIGGER SC_T
AFTER UPDATE OF Grade ON SC
REFERENCING
    OLD row AS OldTuple,
    NEW row AS NewTuple
FOR EACH ROW
WHEN (NewTuple.Grade >= 1.1*OldTuple.Grade)
    INSERT INTO SC_U(Sno,Cno,OldGrade,NewGrade)
VALUES(OldTuple.Sno,OldTuple.Cno,OldTuple.Grade,NewTuple.Grade)
```



触发器示例 2

- [例5.22] 将每次对表Student的插入操作所增加的学生个数记录到表StudentInsertLog中。

```
CREATE TRIGGER Student_Count  
AFTER INSERT ON Student  
/*指明触发器激活的时间是在执行INSERT后*/  
REFERENCING  
NEW TABLE AS DELTA  
FOR EACH STATEMENT  
/*语句级触发器, 即执行完INSERT语句后下面的触发动作体才执行一次*/  
    INSERT INTO StudentInsertLog (Numbers)  
    SELECT COUNT(*) FROM DELTA
```



触发器示例 3

- [例5.23] 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4000元，如果低于4000元，自动改为4000元”。

```
CREATE TRIGGER Insert_Or_Update_Sal  
BEFORE INSERT OR UPDATE ON Teacher  
        /*触发事件是插入或更新操作*/  
REFERENCING NEW row AS newTuple  
FOR EACH ROW      /*行级触发器*/  
BEGIN              /*定义触发动作体，是PL/SQL过程块*/  
    IF (newTuple.Job='教授') AND (newTuple.Sal < 4000)  
    THEN newTuple.Sal :=4000;  
    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，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。



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。

