

第五章 数据库完整性



5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名字句

*5.5 域中的完整性限制

5.6 触发器

5.7 小结

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

- 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

5.6 触发器



❖ 5.6.1 定义触发器

❖ 5.6.2 激活触发器

❖ 5.6.3 删除触发器

5.6.1 定义触发器



❖ 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 >

定义触发器（续）



❖ 定义触发器的语法说明：

- 1. 创建者：表的拥有者
- 2. 触发器名
- 3. 表名：触发器的目标表
- 4. 触发事件： INSERT 、 DELETE 、 UPDATE
- 5. 触发器类型
 - 行级触发器（ FOR EACH ROW ）
 - 语句级触发器（ FOR EACH STATEMENT ）

定义触发器（续）



- ❖ 例如，假设在 [例 11] 的 TEACHER 表上创建了一个 AFTER UPDATE 触发器。如果表 TEACHER 有 1000 行，执行如下语句：

UPDATE TEACHER SET Deptno=5;

- 如果该触发器为语句级触发器，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

定义触发器（续）



❖ 6. 触发条件

- 触发条件为真
- 省略 WHEN 触发条件

❖ 7. 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

定义触发器（续）



[例 18] 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 */
  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 < 4000) THEN
      new.Sal :=4000;
    END IF;
  */
END;
```


定义触发器（续）



[例 19] 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
);
```

定义触发器（续）



[例 19]（续）

```
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;
```

定义触发器（续）



[例 19]（续）

```
CREATE TRIGGER Update_Sal
```

```
AFTER UPDATE ON Teacher
```

/* 触发事件是 UPDATE */

```
FOR EACH ROW
```

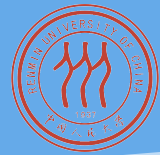
```
AS BEGIN
```

```
IF (new.Sal <> old.Sal) THEN INSERT INTO Sal_log VALUES(  
    new.Eno , new.Sal , CURRENT_USER , CURRENT_TIMESTAMP);
```

```
END IF;
```

```
END;
```

5.6 触发器



❖ 5.6.1 定义触发器

❖ 5.6.2 激活触发器

❖ 5.6.3 删除触发器

5.6.2 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（续）



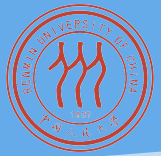
[例 20] 执行修改某个教师工资的 SQL 语句，激活上述定义的触发器。

```
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

5.6 触发器



❖ 5.6.1 定义触发器

❖ 5.6.2 激活触发器

❖ 5.6.3 删除触发器

5.6.3 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 21] 删除教师表 Teacher 上的触发器 Insert_Sal

`DROP TRIGGER Insert_Sal ON Teacher;`

第五章 数据库完整性



5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

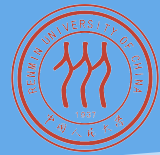
5.4 完整性约束命名字句

*5.5 域中的完整性限制

5.6 触发器

5.7 小结

5.7 小结



- ❖ 数据库的完整性是为了保证数据库中存储的数据是正确的
- ❖ RDBMS 完整性实现的机制
 - 完整性约束定义机制
 - 完整性检查机制
 - 违背完整性约束条件时 RDBMS 应采取的动作