

北京林业大学

数据库原理与应用

触发器



触发器概述

触发器是一种特殊类型的存储过程。

触发器主要有以下优点：

- ◆ 触发器是在某个事件发生时自动激活而执行的。
- ◆ 触发器可以实现比约束更为复杂的完整性要求。
- ◆ 触发器可以根据表数据修改前后的状态采取相应的措施。
- ◆ 触发器可以防止恶意的或错误的INSERT、UPDATE和DELETE操作。



触发器的种类



DML触发器

DML触发器是在执行数据操纵语言 (DML) 事件时被激活而自动执行的触发器。



DDL触发器

DDL触发器是在执行数据定义语言 (DDL) 事件时被激活而自动执行的触发器。



登录触发器

登录触发器是由登录 (LOGON) 事件而激活的触发器。



触发器的工作原理

SQL Server在工作时为每个触发器在服务器的内存上建立两个特殊的表：插入表和删除表。

对表的操作	Inserted表	Deleted表
增加记录 (INSERT)	存放增加的记录	无
删除记录 (DELETE)	无	存放被删除的记录
修改记录 (UPDATE)	存放更新后的记录	存放更新前的记录



触发器的工作原理

(1) INSERT触发器的工作原理

当对表进行INSERT操作时，INSERT触发器被激发，新的数据行被添加到创建触发器的表和Inserted表。



触发器的工作原理

(2) DELETE触发器的工作原理

对表进行DELETE操作时，DELETE触发器被激发，系统从被影响的表中将删除的行放入Deleted表中。



触发器的工作原理

(3) UPDATE触发器的工作原理

当执行UPDATE操作时，UPDATE触发器被激活。触发器将原始行移入Deleted表中，把更新行插入到Inserted表中。



创建触发器

创建DML触发器

使用CREATE TRIGGER创建DML触发器的语法格式为：

CREATE TRIGGER trigger_name

ON {table | view}

[With Encryption]

{For | After | Instead Of}

{ [INSERT] [,] [UPDATE] [,] [DELETE] }

AS sql_statement [;]



创建触发器

创建DML触发器

【例】设计一个触发器，在学生表S中删除某一个学生时，在选课表SC中该学生的选课记录也全部被删除。

```
USE Teach
```

```
GO
```

```
CREATE TRIGGER del_S
```

```
ON S
```

```
AFTER DELETE
```

```
AS
```

```
DELETE FROM SC
```

```
WHERE SC.SNo
```

```
IN (SELECT SNo FROM DELETED)
```

```
GO
```



创建触发器

创建DDL触发器

创建DDL触发器的CREATE TRIGGER语句的语法格式为：

CREATE TRIGGER trigger_name

On {All Server | Database}

[With Encryption]

{FOR | AFTER} {event_type | event_group } [,...n]

AS sql_statement [;]



创建触发器

创建DDL触发器

【例】创建一个DDL触发器safety，禁止修改和删除当前数据库中的任何表。

```
USE Teach
```

```
GO
```

```
CREATE TRIGGER safety
```

```
ON DATABASE
```

```
FOR DROP_TABLE, ALTER_TABLE
```

```
AS PRINT '不能删除或修改数据库表！'
```

```
ROLLBACK
```

```
GO
```



查看触发器

查看表中触发器

执行系统存储过程sp_helptrigger查看表中触发器的语法格式如下：

```
EXEC sp_helptrigger 'table'[, 'type']
```



查看触发器

查看触发器的定义文本

利用系统存储过程sp_helptext可查看某个触发器的内容，语法格式为：

```
EXEC sp_helptext 'trigger_name'
```




查看触发器

查看触发器的所有者和创建时间

系统存储过程sp_help可用于查看触发器的所有者和创建日期，语法格式如下：

```
EXEC sp_help 'trigger_name'
```



修改触发器

利用ALTER TRIGGER语句修改触发器

(1) 修改DML触发器的语法格式如下:

ALTER TRIGGER schema_name.trigger_name

ON (table | view)

[WITH ENCRYPTION]

{ FOR | AFTER | INSTEAD OF }

{ [DELETE] [,] [INSERT] [,] [UPDATE] }

AS sql_statement [;]



修改触发器

利用ALTER TRIGGER语句修改触发器

(2) 修改DDL触发器的语法格式如下:

```
ALTER TRIGGER trigger_name  
ON { ALL SERVER | DATABASE }  
[ WITH ENCRYPTION ]  
{ FOR | AFTER } { event_type | event_group } [ ,...n ]  
AS sql_statement [ ; ]
```




修改触发器

使触发器无效

在有些情况下，用户希望暂停触发器的作用，但并不删除它，这时就可以通过DISABLE TRIGGER语句使触发器无效，语法格式如下：

DISABLE TRIGGER { [schema.] trigger_name [,...n]

| ALL }

ON object_name



修改触发器

使触发器重新有效

要使触发器重新有效，可使用ENABLE TRIGGER语句，语法格式如下：

ENABLE TRIGGER {[schema_name.] trigger_name

[,...n] | ALL }

ON object name



删除触发器

使用DROP TRIGGER语句删除触发器

删除DML触发器的DROP TRIGGER语句的语法格式为：

```
DROP TRIGGER trigger_name [ ,...n ] [ ; ]
```