

## Практическая работа 24

Тема: Триггеры для операций INSERT, UPDATE, DELETE.

### Краткий теоретический материал

Триггеры представляют специальный тип хранимой процедуры, которая вызывается автоматически при выполнении определенного действия над таблицей или представлением, в частности, при добавлении, изменении или удалении данных, то есть при выполнении команд INSERT, UPDATE, DELETE.

Формальное определение триггера:

```
CREATE TRIGGER имя_триггера  
ON {имя_таблицы | имя_представления}  
{AFTER | INSTEAD OF} [INSERT | UPDATE | DELETE]  
AS выражения_sql
```

Для создания триггера применяется выражение CREATE TRIGGER, после которого идет имя триггера. Как правило, имя триггера отражает тип операций и имя таблицы, над которой производится операция.

Каждый триггер ассоциируется с определенной таблицей или представлением, имя которых указывается после слова ON.

Затем устанавливается тип триггера. Мы можем использовать один из двух типов:

- AFTER: выполняется после выполнения действия. Определяется только для таблиц.

- INSTEAD OF: выполняется вместо действия (то есть по сути действие - добавление, изменение или удаление - вообще не выполняется). Определяется для таблиц и представлений.

После типа триггера идет указание операции, для которой определяется триггер: INSERT, UPDATE или DELETE.

Для триггера AFTER можно применять сразу для нескольких действий, например, UPDATE и INSERT. В этом случае операции указываются через запятую. Для триггера INSTEAD OF можно определить только одно действие.

И затем после слова AS идет набор выражений SQL, которые собственно и составляют тело триггера.

Для рассмотрения операций с триггерами определим следующую базу данных productsdb:

```
CREATE DATABASE productsdb;  
GO  
USE productsdb;  
CREATE TABLE Products  
(  
    Id INT IDENTITY PRIMARY KEY,  
    ProductName NVARCHAR(30) NOT NULL,  
    Manufacturer NVARCHAR(20) NOT NULL,  
    ProductCount INT DEFAULT 0,
```

```

    Price MONEY NOT NULL
);
CREATE TABLE History
(
    Id INT IDENTITY PRIMARY KEY,
    ProductId INT NOT NULL,
    Operation NVARCHAR(200) NOT NULL,
    CreateAt DATETIME NOT NULL DEFAULT GETDATE(),
);

```

Здесь определены две таблиц: Products - для хранения товаров и History - для хранения истории операций с товарами.

При добавлении данных (при выполнении команды INSERT) в триггере мы можем получить добавленные данные из виртуальной таблицы INSERTED.

Определим триггер, который будет срабатывать после добавления:

```

USE productsdb
GO
CREATE TRIGGER Products_INSERT
ON Products
AFTER INSERT
AS
INSERT INTO History (ProductId, Operation)
SELECT Id, 'Добавлен товар ' + ProductName + ' фирма ' + Manufacturer
FROM INSERTED

```

Этот триггер будет добавлять в таблицу History данные о добавлении товара, которые берутся из виртуальной таблицы INSERTED.

Выполним добавление данных в Products и получим данные из таблицы History:

```

USE productsdb;
INSERT INTO Products (ProductName, Manufacturer, ProductCount, Price)
VALUES('Lada Vesta', 'BA3', 2, 1500000)
SELECT * FROM History

```

При удалении все удаленные данные помещаются в виртуальную таблицу DELETED:

```

USE productsdb
GO
CREATE TRIGGER Products_DELETE
ON Products
AFTER DELETE
AS
INSERT INTO History (ProductId, Operation)
SELECT Id, 'Удален товар ' + ProductName + ' фирма ' + Manufacturer
FROM DELETED

```

Здесь, как и в случае с предыдущим триггером, помещаем информацию об удаленных товарах в таблицу History.

Выполним команду на удаление:

```
USE productsdb;  
DELETE FROM Products  
WHERE Id=2  
SELECT * FROM History
```

Триггер обновления данных срабатывает при выполнении операции UPDATE. И в таком триггере мы можем использовать две виртуальных таблицы. Таблица INSERTED хранит значения строк после обновления, а таблица DELETED хранит те же строки, но до обновления.

Создадим триггер обновления:

```
USE productsdb  
GO  
CREATE TRIGGER Products_UPDATE  
ON Products  
AFTER UPDATE  
AS  
INSERT INTO History (ProductId, Operation)  
SELECT Id, 'Обновлен товар ' + ProductName + ' фирма ' + Manufacturer  
FROM INSERTED
```

### **Задание**

Создайте три триггера: INSERT, UPDATE, DELETE. Проверьте их работоспособность.

Варианты заданий:

**Вариант 1.** БД «студенческая картотека».

**Вариант 2.** БД расчета зарплаты с окладами, премиями и налогами.

**Вариант 3.** БД учета материалов (приход, расход, остатки, печать накладных).

**Вариант 4.** БД учета платежей за эл. энергию с учетом задолжников.

**Вариант 5.** БД учета платежей за услуги воды и канализации с учетом нормативного потребления.

**Вариант 6.** БД учета кадров на предприятии.

**Вариант 7.** БД «библиотечный каталог».

**Вариант 8.** БД учета успеваемости студентов.

**Вариант 9.** БД учета автотранспорта на предприятии.

**Вариант 10.** БД магазина автозапчастей.

**Вариант 11.** БД станции технического обслуживания автомобилей.

**Вариант 12.** БД «школьный журнал».

В отчете представить текст запросов, а также скриншоты с результатами их выполнения.