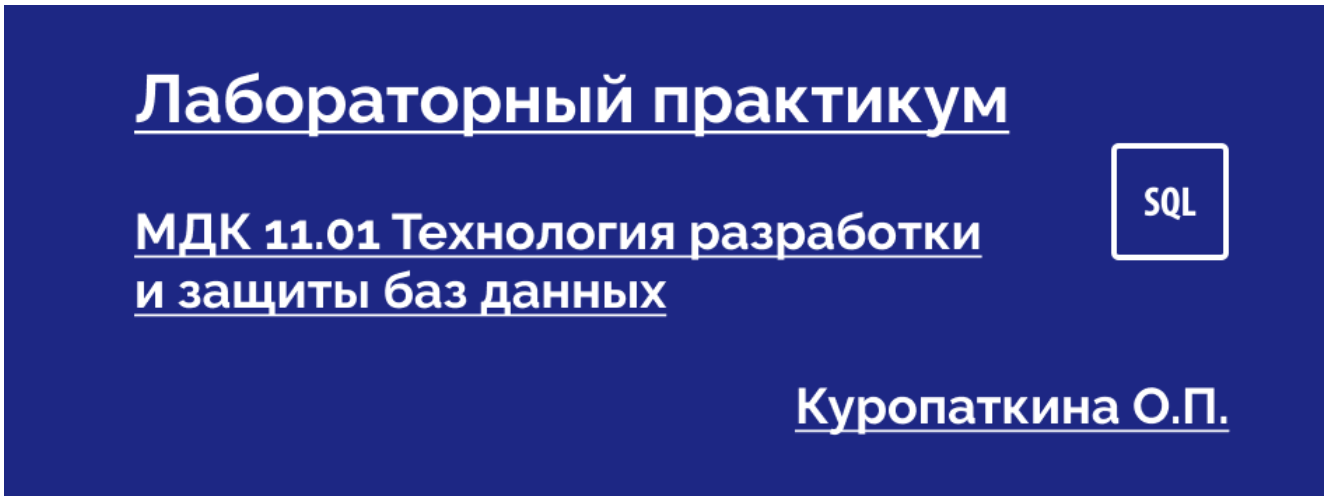

ЛАБОРАТОРНАЯ РАБОТА №13

ТРИГГЕРЫ

Триггер — это особая разновидность хранимой процедуры, которая **автоматически выполняется при возникновении события на сервере базы данных**. Триггеры DML выполняются, когда пользователь пытается изменить данные с помощью событий языка обработки данных (DML). Событиями DML являются процедуры **INSERT, UPDATE или DELETE**, **применяемые к таблице или представлению**. Эти триггеры срабатывают при запуске любого допустимого события независимо от наличия и числа затронутых строк таблицы.

Триггеры – особый инструмент SQL-сервера, используемый для поддержания целостности данных в базе данных. С помощью ограничений целостности, правил и значений по умолчанию не всегда можно добиться нужного уровня функциональности. Часто требуется реализовать сложные алгоритмы проверки данных, гарантирующие их достоверность и реальность. Кроме того, иногда необходимо отслеживать изменения значений таблицы, чтобы нужным образом изменить связанные данные. Триггеры можно рассматривать как своего рода фильтры, вступающие в действие после выполнения всех операций в соответствии с правилами, стандартными значениями и т.д.

Каждый триггер привязывается к конкретной таблице. **Все производимые им модификации данных рассматриваются как одна транзакция**. **В случае обнаружения ошибки или нарушения целостности данных происходит откат этой транзакции**. Тем самым внесение изменений запрещается. Отменяются также все изменения, уже сделанные триггером.

Триггер представляет собой весьма полезное и в то же время опасное средство. Так, при неправильной логике его работы можно легко уничтожить целую базу данных, поэтому триггеры необходимо очень тщательно отлаживать.

В отличие от обычной подпрограммы, триггер выполняется неявно в каждом случае возникновения триггерного события, к тому же он не имеет

аргументов. Приведение его в действие иногда называют запуском триггера. С помощью триггеров достигаются следующие цели:

• проверка корректности введенных данных и выполнение сложных ограничений целостности данных, которые трудно, если вообще возможно, поддерживать с помощью ограничений целостности, установленных для таблицы;

• выдача предупреждений, напоминающих о необходимости выполнения некоторых действий при обновлении таблицы, реализованном определенным образом;

• накопление аудиторской информации посредством фиксации сведений о внесенных изменениях и тех лицах, которые их выполнили;

• поддержка репликации.

Ниже приведен синтаксис создания триггера:

```
CREATE [OR ALTER] TRIGGER [schema_name.]trigger_name -- НАЗВАНИЕ ТРИГГЕРА
ON { table | view }   -- НАЗВАНИЕ ТАБЛИЦЫ/ПРЕДСТАВЛЕНИЯ
{ FOR | AFTER | INSTEAD OF }   -- ВРЕМЯ СРАБАТЫВАНИЯ ТРИГГЕРА
{ [ INSERT ] , [ UPDATE ] , [ DELETE ] }   -- СОБЫТИЕ
AS { sql_statement  [ ; ] [ ,...n ]  }  -- ВЫРАЖЕНИЕ, КОТОРОЕ
                                        -- ВЫПОЛНЯЕТСЯ ПРИ СРАБАТЫВАНИИ ТРИГГЕРА
```

Поподробнее о времени срабатывания триггера.

FOR | AFTER

Значение FOR или AFTER указывает, что триггер DML срабатывает только после успешного запуска всех операций в инструкции SQL, по которой срабатывает триггер. Кроме того, до запуска триггера должны успешно завершиться все каскадные действия и проверки ограничений, на которые есть ссылки.

Нельзя определить триггеры AFTER для представлений.

INSTEAD OF

Указывает, что триггер DML выполняется вместо инструкции SQL, по которой он срабатывает, то есть переопределяет действия запускающих

инструкций. Аргумент INSTEAD OF нельзя использовать для триггеров DDL или триггеров входа.

Для каждой инструкции INSERT, UPDATE или DELETE в таблице или представлении можно определить не более одного триггера INSTEAD OF.

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

Определяет инструкции изменения данных, при применении которых к таблице или представлению срабатывает триггер DML. **Укажите хотя бы один вариант. В определении триггера разрешены любые сочетания вариантов в любом порядке.**

Для триггеров INSTEAD OF нельзя использовать параметр DELETE в таблицах со ссылочной связью, которая определяет каскадное действие ON DELETE. Аналогично параметр UPDATE недопустим в таблицах, у которых есть ссылочная связь с каскадным действием ON UPDATE.

Рассмотрим несколько примеров создания триггеров для существующей базы данных.

Данный триггер срабатывает на обновление и добавление данных в таблицу клиентов и выводит тестовое сообщение.

```
USE ForEducationSQL
GO

CREATE TRIGGER FOREDUCATION_TRIGGER
ON CLIENT -- К ТАБЛИЦЕ 'CLIENT'
FOR INSERT, UPDATE -- СРАБАТЫВАЕТ НА ОБНОВЛЕНИЕ И ДОБАВЛЕНИЕ
AS
BEGIN
    PRINT 'ТРИГГЕР СРАБОТАЛ! ВСЕ ОК!'
END
```

Тестируем работу триггера при обновлении таблицы.

```
12  UPDATE Client
13  SET Phone = '+7(999)999-99-99'
14  WHERE ID = 1
```

```
143 %
Сообщения
ТРИГГЕР СРАБОТАЛ! ВСЕ ОК!

(затронута одна строка)
```

ИСПОЛЬЗОВАНИЕ ТАБЛИЦ INSERTED И DELETED

Инструкции триггеров DML используют **две особые таблицы: deleted и inserted**. SQL Server автоматически создает эти таблицы и управляет ими. Эти **временные таблицы, находящиеся в оперативной памяти, используются для проверки результатов изменений данных и для установки условий срабатывания триггеров** DML. Нельзя в этих таблицах изменять данные напрямую или выполнять над ними операции языка описания данных DDL.

В таблице deleted находятся копии строк, с которыми работали инструкции DELETE или UPDATE. При выполнении инструкции DELETE или UPDATE происходит удаление строк из таблицы триггера и их перенос в таблицу deleted. Таблица триггеров — это таблица, в которой выполняется триггер DML. У таблицы deleted обычно нет общих строк с таблицей триггера.

В таблице inserted находятся копии строк, с которыми работали инструкции INSERT или UPDATE. При выполнении транзакции вставки или обновления происходит одновременное добавление строк в таблицу триггера и в таблицу inserted. Строки таблицы inserted являются копиями новых строк таблицы триггера.

Транзакция обновления аналогична выполнению операции удаления с последующим выполнением операции вставки; сначала старые строки копируются в таблицу deleted, а затем новые строки копируются в таблицу триггера и в таблицу inserted.

Изменим триггер. Добавим условие, если выполнена инструкция **добавления данных в таблицу**, выведем сообщение «Запись в таблицу добавлена!», если **удаления** – «Запись успешно удалена!».

```
ALTER TRIGGER FOREDUCATION_TRIGGER
ON CLIENT -- К ТАБЛИЦЕ 'CLIENT'
FOR INSERT, DELETE -- СРАБАТЫВАЕТ НА ДОБАВЛЕНИЕ И УДАЛЕНИЕ
AS
BEGIN
    IF EXISTS (SELECT * FROM inserted)
    PRINT 'Запись в таблицу добавлена!'
        IF EXISTS (SELECT * FROM deleted)
        PRINT 'Запись успешно удалена!'
END
```

Проверяем работу триггера:

```
20  INSERT INTO Client (FirstName, LastName, Birthday, Email, Phone, GenderCode)
21  VALUES ('Иванов','Иван','16.02.2000','ivanov@mail.ru','+7(999)999-99-98','м')
```

143 %

Сообщения

```
Запись в таблицу добавлена!

(затронута одна строка)
```

```
23  DELETE FROM Client
24  WHERE Phone = '+7(999)999-99-98'
```

143 %

Сообщения

```
Запись успешно удалена!

(затронута одна строка)
```

Рассмотрим несколько примеров триггеров к учебной базе данных Microsoft «AdventureWorks2019».

```
31  CREATE TRIGGER Sales.SaleaOrderDetailNotDiscontinued
32      ON Sales.SalesOrderDetail
33      FOR INSERT, UPDATE
34  AS
35      IF EXISTS
36      (
37          SELECT *
38          FROM Inserted as i -- Данные вставляемые пользователем в таблицу Sales.SalesOrderDetail
39          JOIN Production.Product p
40          ON i.ProductID = p.ProductID
41            WHERE p.DiscontinuedDate IS NOT NULL -- проверяем есть ли дата снятия с продажи
42      )
43  BEGIN
44      PRINT 'Товар отсутсвует на складе!' -- выводим ошибку, если товар уже снят с продажи
45      ROLLBACK TRAN
46  END
47  GO
```

Данный триггер проверяет есть ли дата снятия с продажи в таблице товаров (то есть товар не актуален) и откатывает транзакцию в случае, если товар снят с продажи.

Проверяем работу триггера: Убедимся, что товар с ID = 706 не снят с продажи.

```
49  SELECT * FROM Production.Product WHERE ProductID = 706;
```

143 %

Результаты | Сообщения

	Weight	DaysToManufacture	ProductLine	Class	Style	ProductSubcategoryID	ProductModelID	SellStartDate	SellEndDate	DiscontinuedDate	rowguid	ModifiedDate
1	2.24	1	R	H	U	14	6	2008-04-30 00:00:00.000	NULL	NULL	9540FF17-2712-4C90-A3D1-8CE5568B2462	2014-02-08 10:01:36.827

Осуществим добавление товара в продажу, т.е. добавление товара в таблицу Sales.SalesOrderDetail. Триггер сработать должен.

```
62  INSERT INTO Sales.SalesOrderDetail -- попытаемся добавить этот заказ с товаром ID которого 706
63  VALUES
64  (43659,'4911-403C-98',1,706,1,6.45,0.00,NEWID(),GETDATE());
```

143 %

Сообщения

```
(затронута одна строка)
```

Триггер не сработал, продажа зафиксирована. Все верно.

Теперь присвоим товару с ID = 940 дату снятия с продажи и попытаемся добавить его в продажу.

```
50  UPDATE Production.Product -- создаем проверочную строку где указана дата снятия с продажи
51  SET DiscontinuedDate = '01.01.2021'
52  WHERE ProductID = 940;
53
54  SELECT * FROM Production.Product
55  WHERE ProductID = 940;
```

143 %

Результаты | Сообщения

	DaysToManufacture	ProductLine	Class	Style	ProductSubcategoryID	ProductModelID	SellStartDate	SellEndDate	DiscontinuedDate	rowguid	ModifiedDate
1	1	R	H	NULL	13	70	2013-05-30 00:00:00.000	NULL	2021-01-01 00:00:00.000	44E96967-AB99-41ED-8B41-5BC70A5CA1A9	2014-02-08 10:03:55.510

```
57  INSERT INTO Sales.SalesOrderDetail -- попытаемся добавить этот заказ с товаром ID которого 940
58  VALUES
59  (43659,'4911-403C-98',1,940,1,6.45,0.00,NEWID(),GETDATE());
60
```

143 %

Сообщения

```
The product is out of stock!
Сообщение 3609, уровень 16, состояние 1, строка 57
Транзакция завершилась в триггере. Выполнение пакета прервано.
```

ПРИМЕЧАНИЕ: триггер можно временно отключить без удаления (удаляем как обычно при помощи DROP) с помощью инструкции **DISABLE TRIGGER**

```
-----------------------------------------------------------------------
---отмена действия тригеров на таблице без его удаления---
ALTER TABLE Production.ProductInventory
DISABLE TRIGGER ALL ; -- отключаем все тригеры для указаной таблицы
                      --(либо указываем имя конкретного триггера)
GO
```

Включаем при помощи инструкции **ENABLE TRIGGER**.

САМОСТОЯТЕЛЬНАЯ РАБОТА

1. Создайте триггер, который будет запрещать продажу товара (откатывать транзакцию и выдавать соответствующее сообщение), статус которого - Не для продажи (поле isActive – false).
2. Создайте триггер, который будет при добавлении нового клиента в таблицу Client добавлять в таблицу TegOfClient сведения о присвоении тега добавленному клиенту «Новый клиент».

3. Создайте триггер, который при добавлении записи об оказании услуги в таблицу ClientService будет в зависимости от условия (количество оказанных услуг клиенту превысило пять) добавлять тег «Постоянный клиент».
4. Для дальнейшей работы внесем некоторые изменения в проект базы данных. В таблицу Product добавьте столбец QuantityInStock (Остаток на складе), присвойте значение по умолчанию = 0.

   Создайте триггер, который при добавлении новой продажи товара в таблицу ProductSale будет проверять остаток товара на складе в таблице Product и откатывать транзакцию в случае, если товара недостаточно и выводить соответствующее сообщение. В случае успешного добавления продажи товара, необходимо уменьшать остаток на складе на количество проданного.
5. Создайте триггер, который будет запрещать добавлять запись о продаже товара с неверно указанным количеством (меньше либо равен нулю). Откатите транзакцию и выведите соответствующее сообщение.
6. Создайте триггер, который запретит добавлять в таблицу клиентов младше 16 лет. Откатите транзакцию и выведите соответствующее сообщение.
7. В предыдущей лабораторной работе вы создали хранимую процедуру, которая в зависимости от общей суммы от оказанных услуг клиенту присваивала столбцу PersonalDiscount значение скидки при вызове процедуры.

   Теперь ваша задача создать триггер с аналогичным функционалом. То есть срабатывать триггер должен при добавлении и обновлении записи таблицы ClientService. И обновлять поле PersonalDiscount в зависимости от общей суммы от оказанных услуг клиенту. Если сумма в размере от 0 до 5000 рублей включительно – скидка 5% (0,05), от 5000 до 10000 включительно – скидка 10% (0,1), свыше 10000 – скидка 15% (0,15).