

Учебно-методический материал по теме «Разработка бизнес-логики серверной части приложения»

Специальность 09.02.07 «Информационные системы и программирование»

МДК 11.01 Технология разработки и защиты баз данных

Лабораторная работа Создание триггеров в SQL Server

Цель работы:

- Знакомство с особенностями операторов T- SQL;
- Формирование навыков создания триггеров;
- Формирование навыков использования конструктора среды SQL Server для создания триггеров.

Квалификационные требования:

- Знать назначение и особенности использования триггеров в SQL Server;
- Знать возможности среды SQL Server Management Studio для создания триггеров;
- Уметь создавать триггеры средствами языка T-SQL;
- Уметь сохранить созданную БД в виде сценария и в виде резервной копии.

Ход выполнения:

1. Изучить методические рекомендации;
2. Выполнить по образцу предложенные задачи по созданию триггеров базы данных Банк;
3. Выполнить самостоятельные практические задания;
4. Сохранить созданную БД в виде сценария и в виде резервной копии.



Методические рекомендации к выполнению лабораторной работы

Триггер – это специализированная хранимая процедура, которая может выполняться или для модификации данных, это триггер DML, или для модификации структуры данных, это триггер DDL. Триггеры DML представляют собой фрагменты кода, прикрепленные к определенной таблице и настроенные на автоматическое выполнение в ответ на инструкции INSERT, DELETE, UPDATE. Триггер DDL, напротив, прикреплен к действию, которое происходит либо в базе данных, либо на сервере.

Преимущество триггерной процедуры состоит в том, что SQL Server автоматически вызывает ее при обращении к команде, для которой она определена. В отличие от хранимых процедур триггер невозможно запустить вручную, с триггерами нельзя использовать параметры, и из триггеров невозможно вернуть значения.

Триггеры DML могут служить формой сложных проверок данных, которых невозможно добиться с помощью ограничений. Помните, что использование ограничений вместо триггеров дает выигрыш в производительности. Например, при попытке клиентом совершить транзакцию – покупку акций, соответствующий триггер проверяет, имеется ли у клиента достаточная сумма, а в системе – достаточное количество акций.

Еще одно применение триггера DML - то внесение изменений в другую таблицу на основе того, что произошло в исходной таблице. Например, при совершении транзакции меняется соответствующий баланс клиента.

Для любого действия с таблицами (или комбинации действий) - INSERT, DELETE, UPDATE, за исключением SELECT, можно создать триггеры. Очевидно, что при выполнении инструкции SELECT в таблице не происходят никакие изменения, поэтому и создать триггер невозможно.

SQL Server поддерживает два различных типа триггеров: триггеры AFTER и триггеры INSTEAD OF.

Триггеры AFTER вызываются после выполнения команды, которой они назначены. Триггеры AFTER вы можете использовать для команд INSERT, UPDATE и DELETE. Триггеры AFTER можно создавать только для таблиц, но не для представлений. Для каждой из этих трех команд может быть установлено несколько триггеров.

Триггеры INSTEAD OF заменяют команду, для которой они объявлены. Подобно триггерам AFTER, вы можете определять триггеры INSTEAD OF для команд INSERT, UPDATE или DELETE. Один триггер может быть применен к нескольким командам. Однако, в отличие от триггеров AFTER, вы можете создавать триггеры INSTEAD OF как для таблиц, так и для представлений, но для каждого действия над этой таблицей или представлением может быть создан только один триггер INSTEAD OF.

Базовый синтаксис оператора создания триггера - следующий:

```
CREATE TRIGGER имя_триггера
ON таблица_или_представление
тип_триггера, список_команд
AS
оператор_SQL
```

Таблица_или_представление может быть именем единственной таблицы или представления, с которой связан триггер. Представление возможно только если тип_триггера есть INSTEAD OF, поскольку только этот тип триггера вы можете определять для представлений. Триггеры не могут быть созданы для временных или системных таблиц, но они могут ссылаться на временные таблицы. Тип_триггера должен быть выражен одним из ключевых слов AFTER, FOR или INSTEAD OF, в то время как список команд может быть любой комбинацией команд INSERT, UPDATE или DELETE. Если вы указываете более одной команды, их следует отделять запятыми.

При выполнении заданий используется **база данных Банк**.



Восстановить базу данных Банк версия 04 из скрипта или резервной копии



Создание триггера по образцу

Создадим триггер, который выполняется при изменении поля Фамилия таблицы Клиенты и сохраняет информацию о старой фамилии в таблице ЖурналИзмененияФамилий с полями КодКлиента, ПрежняяФамилия, ДатаИзменения.

1. Откройте окно Создать Запрос. В появившемся окне введите приведенный ниже код.

```
USE [Банк]
GO
CREATE TABLE ЖурналИзмененияФамилий
( КодКлиента BIGINT NOT NULL,
  ПрежняяФамилия NVARCHAR(50) NOT NULL,
  ДатаИзменения DateTime NOT NULL,
)
GO
CREATE TRIGGER [dbo].[trg ИзменилиФамилиюУКлиента]
ON Клиенты
AFTER UPDATE
AS
IF UPDATE (Фамилия)
INSERT INTO ЖурналИзмененияФамилий (КодКлиента, ПрежняяФамилия,
ДатаИзменения)
SELECT D.КодКлиента, D.Фамилия, GetDate()
```

```
FROM Клиенты INNER JOIN DELETED D
ON Клиенты.КодКлиента=D.КодКлиента
```

GO

- Код триггера выполняется автоматически после изменения фамилии. Добавьте в таблицу Клиенты нового клиента с фамилией Сидоров. Затем, измените его фамилию на Сидоренко. После сохранения изменений и обновления таблиц убедитесь, что в ЖурналеИзмененияФамилий появилась соответствующая запись. (рис.1)

	КодКлиента	ПрежняяФами...	ДатаИзменения
▶	9	Сидоров	2017-06-09 11:...
*	NULL	NULL	NULL

Рис. 1 Сохранение изменений в журнале



Создание триггера по образцу

Для таблицы ФинансовыеОперацииКлиентов создадим триггер, который будет менять текущее сальдо счета клиента после вставки новой операции этого клиента. Для расчета нового остатка используется формула: новый остаток = старый остаток счета + (или минус) сумма транзакции (в зависимости от типа транзакции: депозит или снятие).

Данный триггер будет триггером AFTER для инструкции INSERT таблицы dbo.ФинансОперацииКлиентов.

- Откройте таблицу dbo.ФинансовыеОперацииКлиентов в дереве объектов БД Банк, на ее папке Триггеры из правого контекстного меню вызовите команду Создать Триггер. В появившемся шаблоне проведите необходимые изменения, чтобы получился приведенный ниже код.

```
USE [Банк]
```

```
GO
```

```
CREATE TRIGGER [dbo].[trgИзменБалансаПослеНовТранзак]
ON [dbo].[ФинансовыеОперацииКлиентов]
AFTER INSERT
AS
```

```
Declare @СуммаОпер Numeric(18,5),
        @ТекСальдо money
```

```
SELECT
```

```
@ТекСальдо=k.ТекущСальдоСчета,
```

```
@СуммаОпер=
```

```
CASE
```

```
WHEN (dbo.ТипыОпераций.ТипКредитаДебета=1) AND
      (dbo.ТипыОпераций.ВлияниеОперацииНаБаланс=1)
```

```
THEN I.Сумма
```

```
WHEN (dbo.ТипыОпераций.ТипКредитаДебета=0) AND
      (dbo.ТипыОпераций.ВлияниеОперацииНаБаланс=1)
```

```
THEN (-1)*I.Сумма
```

```
END
```

```
FROM      (dbo.Клиенты AS K INNER JOIN INSERTED AS I
```

```
ON K.КодКлиента=I.КодКлиента) INNER JOIN dbo.ТипыОпераций
```

```
ON I.ТипОперации = dbo.ТипыОпераций. КодТипа
```

```
UPDATE dbo.Клиенты
```

МДК11.01 "Технология разработки и защиты баз данных"

```
SET ТекущСальдоСчета=@ТекСальдо +@СуммаОпер
```

```
FROM Клиенты INNER JOIN INSERTED I on Клиенты.КодКлиента=I.КодКлиента
```

```
GO
```

В данном коде вводятся локальные переменные для суммы операции и хранения старого остатка. Затем проводятся расчеты суммы зачисления, в зависимости от типа операции: дебетовая - сумма зачисляется с плюсом, кредитовая - с минусом. После проведенных расчетов проводится обновление таблицы Клиенты с вычислением нового остатка счета клиента, совершившего транзакцию.

Чтобы определить, для какого клиента следует изменить текущий остаток, используется таблица INSERTED, которая содержит строку новой транзакции этого клиента, из нее берется информация о коде этого клиента, типе новой операции и сумме совершенной операции.

2. Код триггера выполняется автоматически после ввода новой транзакции. Введите новую транзакцию для любого клиента, затем убедитесь, что его текущее сальдо счета пересчитывается правильно. (рис.2, рис.3)

КодОперации	ТипОперации	ДатаОперации	Сумма	КодКлиента
1	1	2017-06-13 14:...	1000,00000	4
2	2	2017-06-13 14:...	4000,00000	6
3	1	2017-06-13 00:...	50000,00000	6
4	3	2017-06-13 16:...	10000,00000	7
10	1	2017-06-13 00:...	5000,00000	8

Рис. 2 Ввод новой транзакции для клиента с кодом=8 и предыдущим сальдо=0 р.

КодКлиента	Фамилия	ДрИнициалы	Имя	Адрес	ИдентБанкСчета	ТекущСальдо...	ДатаОткСчета
4	Петров	NULL	Иван	Новосибирск	98776556678	1000,0000	2012-04-06
6	Козлов	NULL	Семен	Омск	765398087	18000,0000	2012-04-10
7	Иванов	NULL	Петр	Омск	09876098	50000,0000	2012-04-09
8	Иванов	NULL	Олег	Новосибирск	5643987	5000,0000	2010-04-09

Рис. 3 Изменение баланса для клиента с кодом=8

Замечание. Обязательно сохраните изменения и обновите таблицы.



Практические задания



Самостоятельное создание триггеров для базы Банк

1. Создайте новую таблицу ДатыПоследнихТранзакцийКлиентов, содержащую КодКлиента, Фамилию, Имя, НомерСчета, ДатуПоследнОперации. Создайте триггер, который выполняется после вставки транзакции в таблицу ФинансовыеОперацииКлиентов и сохраняет дату этой операции в поле ДатаПоследнОперации таблицы ДатыПоследнихТранзакцийКлиентов. Проверьте его работу.

Рекомендации к выполнению: использовать таблицу INSERTED.

2. Создайте таблицу, содержащую столбцы с Фамилией и Именем Клиента, а также с его текущим сальдо счета. Добавьте также в новую таблицу поле Удален типа Bit, значение по умолчанию которого, равно нулю. Заполните новую таблицу клиентами с нулевым текущим сальдо счета. Создайте триггер, который выполняется вместо удаления клиента и заполняет поле Del значением 1. Проверьте его работу.

МДК11.01 "Технология разработки и защиты баз данных"

Рекомендации к выполнению: используйте таблицу DELETED для получения ключевого поля удаляемой записи.



Сохраните базу Банк в виде сценария и виде резервной копии в свою учебную папку с именем Банк_версия05.