Лабораторная работа № 8

Использование хранимых процедур и триггеров  
 при работе с базами данных

**Цель работы:** изучить и использовать воможности хранимых пороцедур и триггеров.

**Контрольные вопросы**

1. Что представляет собой хранимая процедура?

Хранимая процедура представляет собой оформленный особым образом пакет, который хранится в базе данных. Хранимые процедуры отличаются от пакетов тем, что в них допускается использование входных и выходных параметров, а также возвращаемых значений

1. Как создать хранимую процедуру?

CREATE процедура(@параметры)?

1. Как выполнить хранимую процедуру?
2. Как передать данные в хранимую процедуру?

Через (@параметр1, @параметр2)?

1. Как получить результаты из хранимой процедуры?
2. Каково назначение команды Return?

Возвращает код ошибки?

1. Как получить возвращаемое значение из хранимой процедуры?
2. Что представляет собой триггер?

Триггер — это хранимая процедура особого типа, вызываемая на выполнение в от­вет на определенные события. Триггеры подразделяются на два основных класса: триг­геры языка определения данных (Data Definition Language — DDL) и триггеры языка манипулирования данными (Data Manipulation Language — DML).

1. Перечислите отличия триггера от хранимой процедуры.

В отличие от хранимых процедур, при использовании которых требуется их явный вызов на выполнение, триггеры вызываются на выполнение автоматически при обнаружении события (или событий), связанного с выполнением операций над таблицей, за которой закреплен триггер. Триггеры не могут быть вызваны явно; единственный способ обеспечения вызова состоит в выполнении требуемого действия над табли­цей, за которой закреплен триггер.

Различия между хранимыми процедурами и триггерами не ограничиваются тем, что невоз­можен явный вызов триггера. Хранимые процедуры не только вызываются явно, но и отли­чаются двумя особенностями, которыми не обладают триггеры: (1) принимают и передают параметры и (2) возвращают коды завершения.

Триггеры не имеют параметров, но в коде триггера может использоваться определенный механизм, позволяющий выяснить, на какие строки должно распространяться действие триггера. Еще одна особенность триггеров состоит в том, что в них допускается применение команды RETURN, но они не позволяют возвращать конкретные значения кода завершения (поскольку явный вызов триггера не предусмотрен, то не определена точка вызова, в которую можно было бы возвратить код завершения).

1. Какие классы триггеров предусмотрены?

Триггеры подразделяются на два основных класса: триг­геры языка определения данных (Data Definition Language — DDL) и триггеры языка манипулирования данными (Data Manipulation Language — DML).

1. Какие типы триггеров DML существуют?

-Триггеры INSERT.

-Триггеры DELETE.

-Триггеры UPDATE.

-Триггеры, создаваемые с учетом одновременного возникновения и совпадения событий.

1. Какие подклассы триггеров DML существуют и каковы их особенности?

В СУБД SQL Server существуют два подкласса триггеров DML: INSTEAD OF (триггер замены операции) и AFTER (триггер, выполняющийся сразу после операции), отли­чающиеся своим назначением, моментом выполнения и производимым эффектом (табл. 1).

|  |  |  |
| --- | --- | --- |
| Характеристика | Триггер INSTEAD OF | Триггер AFTER |
| Эффект оператора DML | Автоматически откатывается | Выполняется, если триггер сам не откатит транзакцию |
| Момент выполне-ния триггера | Перед проверкой ограниче-ний первичного и внешнего ключей | После выполнения транзакции, но перед ее подтверждением |
| Количество воз-можных событий таблицы | Одно | Несколько |
| Возможность при-менения к пред-ставлениям | Есть | Отсутствует |
| Рекурсивное срабатывание | Отсутствует | Зависит от параметров настройки СУБД |

1. Как временно отключить срабатывание триггера, а затем снова включить?

Оператор DML не может повлиять на срабатывание триггера, однако триггер можно временно отключить оператором ALTER TABLE с па­раметром DISABLE TRIGGER:

ALTER TABLE *имя\_таблицы* DISABLE TRIGGER *имя\_триггера*

Для включения триггера применяется тот же оператор, но с пара­метром ENABLE TRIGGER:

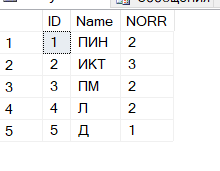
ALTER TABLE *имя\_таблицы* ENABLE TRIGGER *имя\_триггера*

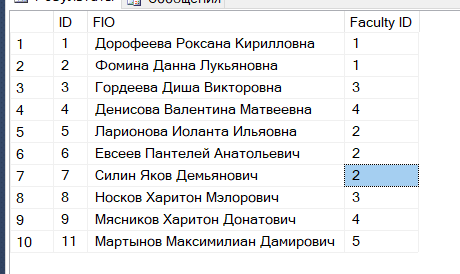
1. Что содержат виртуальные таблицы Inserted и Deleted?

|  |  |  |
| --- | --- | --- |
| Оператор DML | Виртуальная таблица | |
| Inserted | Deleted |
| insert | Вставленные строки | Пустая |
| update | Строки базы данных после обновления | Строки базы данных до обновления |
| delete | Пустая | Строки, подлежащие удалению |

**Задания**

1. Дополнить выбранную главную таблицу столбцом NORR с нулевым значением по умолчанию и заполнить его значениями, равными числу подчиненных строк.





1. Создать хранимые процедуры и убедиться в их наличии в базе данных, выбрав ветви Programming и Stored Procedures в окне Обозревателя объектов

--количество книг студента

CREATE PROCEDURE pGetCountBook(

@StudentID INT,

@CountBook INT OUTPUT

)

AS

SET NOCOUNT ON

SELECT @CountBook= COUNT(ID)

FROM Book

WHERE [Student ID]=@StudentID

--передача книги другому студенту

CREATE PROCEDURE pUpdBook(

@ID INT,

@StudentID INT

)

AS

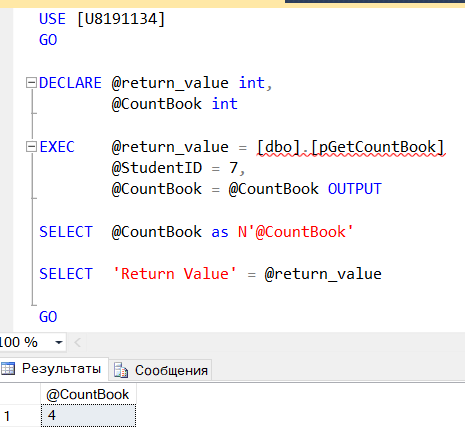
SET NOCOUNT ON

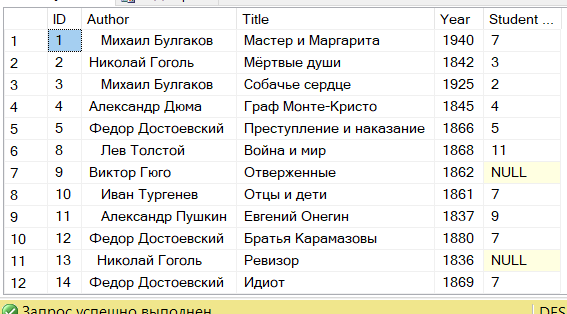
UPDATE Book SET [Student ID] = @StudentID

WHERE ID = @ID

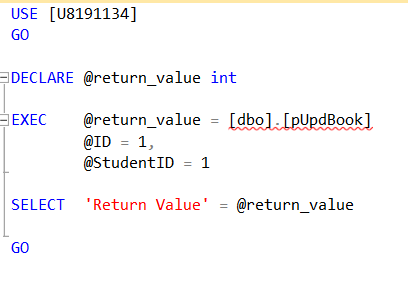
1. Выполнить тестирование хранимых процедур и прокомментировать результаты в отчете.

1)





2)





1. Создать триггеры и убедиться в их наличии в базе данных, выбрав ветвь Triggers для таблиц базы данных в окне Обозревателя объектов.

CREATE TRIGGER update\_book\_trig ON Book

AFTER UPDATE

AS

PRINT('Книга передана другому студенту');

CREATE TRIGGER delete\_book\_trig ON Book

INSTEAD OF DELETE

AS

PRINT 'Невозможно удалить книгу'

1. Выполнить тестирование триггеров и прокомментировать в отчете результаты их действия.

