

```

USE tempdb
-- Создаем во временной БД тестовую таблицу
GO
CREATE TABLE IdentityTest(
    Id int IDENTITY(1,1) NOT NULL,      -- IDENTITY(1,1) - задает автоматическое увеличение значения,
                                        -- начиная с 1 и на 1 шаг при каждой попытке
                                        -- (в т.ч. и неудачной!) вставки новой записи
    [Data] nvarchar(50) NOT NULL
)
GO
-- В простых случаях эти функции возвращают одинаковый ПРАВИЛЬНЫЙ результат
-- - текущее (последнее) значение счетчика идентификатора таблицы
INSERT INTO IdentityTest(Data) VALUES('Первая строка')
SELECT @@IDENTITY, SCOPE_IDENTITY(), IDENT_CURRENT('IdentityTest')
-- Получаем: 1, 1, 1 - Правильно

-- НО! Если у таблицы есть триггер на вставку, который сам выполняет вставку, то возникает путаница с глобальной
переменной @@IDENTITY

-- Создаем еще одну таблицу
GO
CREATE TABLE IdentityTest2(Id int IDENTITY(1,1) NOT NULL, [Data] nvarchar(50) NOT NULL)
GO
-- Создаем триггер для первой таблицы, в которой вставляется запись во вторую таблицу
GO
CREATE TRIGGER ins_IdentityTest ON  dbo.IdentityTest AFTER INSERT AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO IdentityTest2(Data) VALUES('Первая строка')
END
GO

-- Еще раз тестируем функции
INSERT INTO IdentityTest(Data) VALUES('Вторая строка')
SELECT @@IDENTITY, SCOPE_IDENTITY(), IDENT_CURRENT('IdentityTest'), IDENT_CURRENT('IdentityTest2')
-- Получаем: 1 (??? - это счетчик IdentityTest2), 2, 2, 1
-- @@IDENTITY вернул текущее значение счетчика для ЛЮБОЙ ПОСЛЕДНЕЙ таблицы,
-- в которую была выполнена вставка
-- (в данном случае она была выполнена в триггере на вставку для таблицы IdentityTest)
-- Остальные функции вернули верное значение
-- В функции IDENT_CURRENT явно указываем, из какой таблицы нужно прочесть текущее значение счетчика

-- Удаляем триггер, чтобы не путаться со счетчиками между разных таблиц
DROP TRIGGER ins_IdentityTest

INSERT INTO IdentityTest(Data) VALUES('Третья строка')
INSERT INTO IdentityTest2(Data) VALUES('Вторая строка')
SELECT @@IDENTITY, SCOPE_IDENTITY(), IDENT_CURRENT('IdentityTest'), IDENT_CURRENT('IdentityTest2')
-- @@IDENTITY и SCOPE_IDENTITY() возвращают значение для таблицы из последней вставки (IdentityTest2)
-- IDENT_CURRENT('IdentityTest') и IDENT_CURRENT('IdentityTest2') возвращают корректные значения каждая для своей
таблицы

-- НО! Если выполняется вставка нескольких записей, то для получения списка идентификаторов используется
предложение OUTPUT
DECLARE @Identities TABLE(Id INT)

-- Вставляем СПИСОК значений
INSERT INTO IdentityTest(Data)
OUTPUT inserted.Id INTO @Identities
VALUES('Четвертая строка'), ('Пятая строка')

SELECT *, @@IDENTITY, SCOPE_IDENTITY(), IDENT_CURRENT('IdentityTest'), IDENT_CURRENT('IdentityTest2') FROM
@Identities
-- Этот простой и надежный способ получения любых значений из таблицы, в т.ч. установленных по умолчанию или
измененных триггерами

-- Предыдущие функции вернут только самые последние значения счетчиков, а не список
SELECT @@IDENTITY, SCOPE_IDENTITY(), IDENT_CURRENT('IdentityTest'), IDENT_CURRENT('IdentityTest2')

DROP TABLE IdentityTest
DROP TABLE IdentityTest2

```