**Цели работы:**

* научиться заполнять таблицы данными средствами MS SQL Server Management Studio, а также средствами языка T-SQL;
* изучить возможности оператора INSERT INTO для работы с данными.

**Теоретическая часть:**

Для добавления данных применяется команда **INSERT**, которая имеет следующий формальный синтаксис:

INSERT [INTO] {[database\_name.][schema\_name.]table\_name}

[(column\_name [,…n])]

{

VALUES ({NULL| DEFAULT| expression})

| DEFAULT VALUES

| SELECT <select\_criteria>

}

где **INTO** – необязательное ключевое слово, которое можно использовать между ключевым словом INSERT и целевой таблицей.

**database\_name** – имя базы данных.

**column\_list** – список, состоящий из одного или нескольких столбцов (**column\_name**), в которые вставляются данные. Список должен быть заключен в круглые скобки, а его элементы должны разделяться запятыми.

Если столбец не внесен в column\_list, то должно быть обеспечено значение, основанное на определении столбца; в противном случае строку нельзя будет загрузить. Автоматически задаются значения для столбца, если столбец имеет следующие характеристики:

* имеется свойство IDENTITY – используется следующее значение приращения для идентификатора.
* имеется стандартное значение – используется стандартное значение для столбца.
* имеет тип данных timestamp – используется текущее значение отметки времени.
* допускает значение NULL – используется значение NULL.
* вычисляемый столбец – используется вычисленное значение.

Когда в столбец идентификаторов вставляются явно заданные значения, то параметру SET IDENTITY\_INSERT необходимо присвоить значение ON.

Ели необходимо вставить значение идентификатора, используется системная функция @@IDENTITY.

После завершения инструкций INSERT, SELECT INTO или массового копирования функция @@IDENTITY возвращает последнее значение идентификатора, сформированное инструкцией. Если инструкция не обработала ни одной таблицы, содержащей столбцы идентификаторов, функция @@IDENTITY возвращает значение NULL. Когда при вставлении нескольких строк формируется несколько значений идентификаторов, функция @@IDENTITY возвращает последнее сформированное значение.

**VALUES** – позволяет использовать один или несколько списков вставляемых значений данных. Для каждого столбца в column\_list, если этот параметр указан, или в таблице должно быть одно значение. Список значений должен быть заключен в скобки.

Если значения в списке идут в порядке, отличном от порядка следования столбцов в таблице, или не для каждого столбца таблицы определено значение, то необходимо использовать аргумент column\_list для явного указания столбца, в котором хранится каждое входное значение.

**DEFAULT** – указывает на необходимость принудительно загружать значения по умолчанию, определенные для столбца. Если для столбца не задано значение по умолчанию, и он может содержать значение NULL, вставляется значение NULL. В столбцы с типом данных timestamp вставляется следующее значение метки времени. Значение DEFAULT недопустимо для столбца идентификаторов.

Существует несколько способов использования инструкции INSERT в части данных, которые необходимо вставить:

* перечисление конкретных значений для вставки;
* указание набора данных в виде запроса SELECT;
* указание набора данных в виде вызова процедуры, которая возвращает табличные данные.

**Вставка одной строки данных**

Так как значения для всех столбцов предоставлены и перечислены в том же порядке, что и столбцы в таблице, не требуется указывать имена столбцов в списке столбцов.

INSERT INTO Production.UnitMeasure

VALUES (N'FT', N'Feet', '20080414')

**Вставка нескольких строк данных**

Конструктор значений таблицы или строки позволяет вставить в таблицу несколько записей (строк) посредством инструкции языка DML, такой как, например, INSERT.

INSERT INTO Production.UnitMeasure

VALUES (N'FT2', N'Square Feet ', '20080923'),

(N'Y', N'Yards', '20080923'),

(N'Y3', N'Cubic Yards', '20080923')

**Вставка данных в порядке, отличном от порядка столбцов таблицы**

В следующем примере используется список столбцов для явного указания значений, которые будут вставляться в каждый столбец. Порядок следования столбцов в таблице Production.UnitMeasure базы данных AdventureWorks является следующим: UnitMeasureCode, MeasureName, ModifiedDate. Но в column\_list столбцы перечислены в ином порядке.

Для вставки значений даты может использоваться функция GETDATE(), которая возвращает текущую системную метку времени базы данных в виде значения datetime без смещения часового пояса базы данных. Это значение наследуется от операционной системы компьютера, на котором работает экземпляр SQL Server.

INSERT INTO Production.UnitMeasure

(MeasureName, UnitMeasureCode, ModifiedDate)

VALUES (N'Square Yards', N'Y2', GETDATE()

**Вставка данных в таблицу со столбцами, имеющими значение по умолчанию**

Столбцы не указаны, и поэтому вставляются только значения по умолчанию с помощью предложения DEFAULT VALUES (в случае, если все столбцы имеют значение по умолчанию).

INSERT INTO T1

DEFAULT VALUES

**Вставка данных из других таблиц с помощью параметров SELECT и EXECUTE**

**Вставка данных, указав набор данных в виде запроса SELECT**

Инструкция

INSERT INTO <target\_table>

SELECT <columns>

FROM <source\_table>

может эффективно перенести большое количество строк из одной таблицы (например, промежуточной) в другую таблицу с минимальным протоколированием.

INSERT INTO EmployeeTitles

SELECT EmployeeKey, LastName, Title

FROM DimEmployee

WHERE EndDate IS NULL

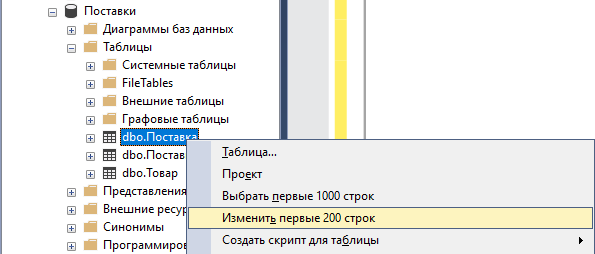
**Вставка данных, используя вызов процедуры, которая возвращает табличные данные**

INSERT INTO dbo.EmployeeSales

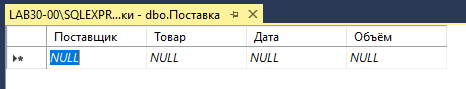
EXECUTE dbo.uspGetEmployeeSales

**Вставка данных, используя графический интерфейс**

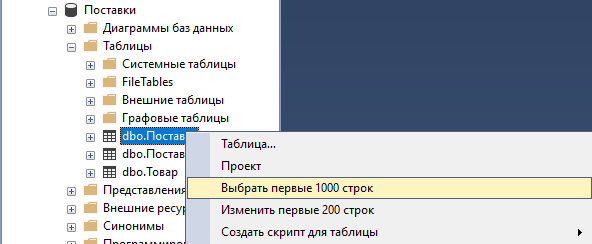
Для этого необходимо выполнить пункт контекстного меню таблицы Изменить первые 200 строк.



После чего заполнить ячейки конкретными значениями.



Просмотреть внесённую информацию можно, выполнив пункт контекстного меню таблицы Выбрать первые 1000 строк.



**Задание:**

* 1. Запустить SQL Server Management Studio, выполнить подключение к серверу (используя параметры соединения, аналогичные практической 1).
  2. Заполнить созданные в практической работе 1 таблицы (по 10 записей в главных таблицах и 15 записей в подчиненной таблице). Для генерации данных можно использовать любой из сервисов (например, <https://randomdatatools.ru>). Данные необходимо предварительно подготовить.

**В отчет предоставить:**

* текст SQL-запроса, выполняющий вставку данных;
* скриншоты содержимого таблиц БД, показывающие добавленные данные (на скриншоте обязательно должно присутствовать имя (номер) компьютера).

**Контрольные вопросы:**

1. Опишите базовый синтаксис оператора INSERT.
2. Для каких столбцов автоматически задаются значения?
3. За что отвечает SET IDENTITY\_INSERT?
4. Для чего предназначена функция @@IDENTITY?
5. Имеет ли значение порядок вставляемых значений? Ответ обоснуйте.
6. Как можно использовать инструкцию INSERT?
7. Для чего предназначена функция GETDATE()?
8. Как осуществить вставку данных в таблицу со столбцами, имеющими значение по умолчанию?
9. Как осуществить вставку данных, указав набор данных в виде запроса SELECT?
10. Как осуществить вставку данных, используя вызов процедуры, которая возвращает табличные данные?

**Порядок оценивания:**

Оценка «**Отлично**» выставляется, если были выполнены все пункты задания, отчёт содержит всю необходимую информацию по проделанной работе, оформлен согласно требованиям, работа сдана и защищена в срок, в ходе защиты даны правильные ответы на 5 вопросов.

Оценка «**Хорошо**» выставляется, если были выполнены все пункты задания, отчёт содержит всю необходимую информацию по проделанной работе, оформлен согласно требованиям, работа сдана и защищена в срок, в ходе защиты даны правильные ответы на 4 вопроса.

Оценка «**Удовлетворительно**» выставляется, если были выполнены все пункты задания, отчёт содержит минимально необходимую информацию по проделанной работе, работа сдана и защищена с нарушением сроков, в ходе защиты даны правильные ответы на 2-3 вопроса.

Оценка «**Неудовлетворительно**» выставляется, если были выполнены не все пункты задания, отчёт содержит информацию, не позволяющую оценить ход выполнения работы, работа сдана с нарушением сроков, не защищена, либо в ходе защиты даны правильные ответы менее чем на 2 вопроса.

Работы, получившие оценку «Удовлетворительно» и выше являются засчитанными.

Работы, получившие оценку «Неудовлетворительно» являются обязательными к доработке и повторной сдаче. Итоговая оценка при повторной сдаче работы будет снижена на 1 балл.