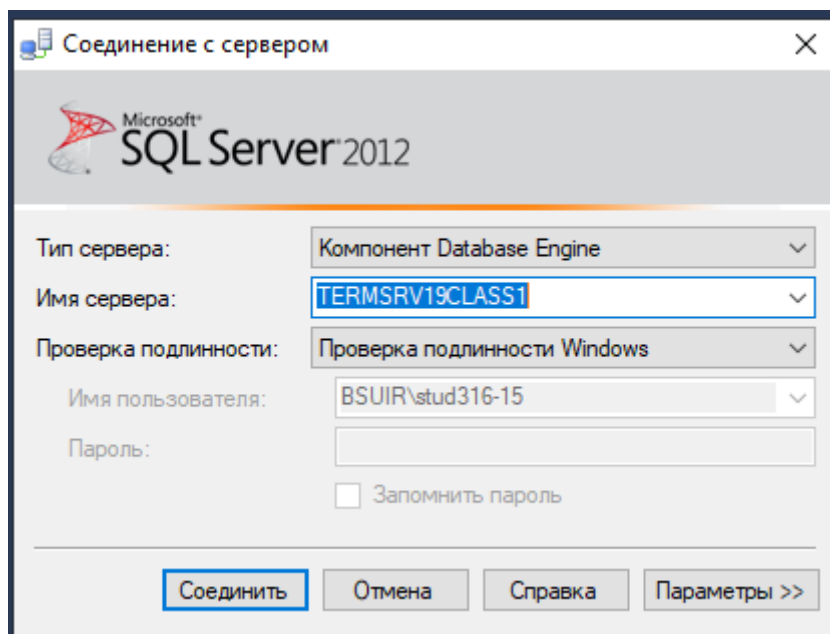


## Лабораторная работа № 9.

### Операции с БД средствами языка манипулирования данными – DML - SQL

Для начала работы запустите среду Microsoft SQL Server Management Studio известным Вам способом и установите соединение с Вашим сервером СУБД. Например:



Для выполнения заданий данной ЛР-9 используйте БД с названием «Номер\_группы\_Фамилия\_ЛР\_8», которая была создана в процессе выполнения Вами заданий предыдущей ЛР-8.

Прочитайте теоретический материал и выполните [задания для самостоятельной работы \(в конце\)](#) с оформлением отчета и обязательными скриншотами результатов.

### ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ ДЛЯ ИЗУЧЕНИЯ

**Инструкция INSERT** позволяет вставлять в таблицу **новые** строки.

В зависимости от схемы таблицы вам может понадобиться предоставить данные для всех или части столбцов таблицы.

Инструкция **INSERT** позволяет вставлять в таблицы одну строку или сразу несколько строк, как **на основе параметров команды**, так и **на основе вложенного запроса на выборку** из другой таблицы.

*У команды INSERT следующая синтаксическая запись:*

*INSERT [ TOP ( expression ) [ PERCENT ] ]*

*[ INTO ]*

*{ <object> | rowset\_function\_limited [ WITH ( <Table\_Hint\_Limited> [ ...n ] ) ] }*

```

{ [ ( column_list ) ] [ <OUTPUT Clause> ]
{ VALUES ( { DEFAULT | NULL | expression } [ ,...n ] ) [ ,...n ]
/ derived_table
/ execute_statement
/ <dml_table_source>
/ DEFAULT VALUES } }
[; ]

```

Вставка в заданную таблицу или представление **новой строки (строк) непосредственно из команды.**

**Синтаксис команды:**

```

INSERT [INTO] имя_таблицы [(список_имен_столбцов)]
VALUES (список_значений) [, (список_значений)]

```

Если **список\_имен\_столбцов** не указан, то будут заполнены все столбцы в порядке, указанном в команде **CREATE TABLE**, иначе требуется указать значения для заполняемых столбцов, в том числе обязательно для имеющих ограничение NOT NULL, остальные поля строки получают значение NULL или DEFAULT.

Если в аргументе **список\_имен\_столбцов** указаны не все столбцы таблицы или представления, то в пропущенные столбцы вставляется либо значение по умолчанию (если для столбца оно задано), или значение NULL.

Для всех пропущенных столбцов должно быть либо определено значение по умолчанию, либо в ограничениях допускаться значение **NULL**.

Этот пример вставляет одну строку в таблицу Production.UnitMeasure.

Так как значения для всех столбцов предоставлены и перечислены в том же порядке, что и столбцы таблицы, имена столбцов не указаны в параметре column\_list.

```

INSERT [INTO] Production.UnitMeasure
VALUES (N'F2', N'Square Feet', GETDATE());

```

Этот пример вставляет несколько строк в таблицу Production.UnitMeasure.

```

INSERT INTO Production.UnitMeasure
VALUES (N'F2', N'Square Feet', DEFAULT), (N'Y2', N'Square Yards', DEFAULT );

```

При желании порядок перечисления значений столбцов можно изменить, но при этом нужно указать их имена во фразе INTO в желаемом порядке.

Вставка **новых строк на основе запроса** по другим таблицам (в сочетании с дополнительным оператором SELECT).

**Синтаксис команды:**

```

INSERT INTO имя_таблицы [(список_имен_столбцов)]
SELECT ...FROM .....(выборка строк из другой таблицы или таблиц)

```

При выполнении команды ограничения проверяются для каждой вставляемой строки и в случае их нарушения хотя бы для одной из строк **все данные не будут вставлены** – автоматически СУБД выполнится команда ROLLBACK – откат.

В данном примере в таблицу оценок будут добавлены все студенты, а в столбец «предметы» будет помещен предмет с кодом 5.

```
INSERT INTO marks (cod_st, cod_sub)
SELECT students.cod_st, subject.cod_sub
FROM students, subject WHERE cod_sub=5
```

**Инструкция DELETE** удаляет одну или несколько строк **целиком** из таблицы или представления.

**Синтаксис команды:**

```
DELETE FROM имя_таблицы | имя_представления
[ WHERE условие отбора строк для удаления ]
или без WHERE - полная очистка таблиц.
```

Например – удалить из таблицы «Студенты» всех студентов с фамилией «Иванов»:

```
DELETE FROM students WHERE name_st='Иванов'
```

Нельзя удалить строку родительской таблицы, если есть связанные строки в подчиненной таблице и действует правило (ограничение) ссылочной целостности ON DELETE RESTRICT.

Если действует правило (ограничение) ссылочной целостности **ON DELETE CASCADE** - будут автоматически удаляться и все связанные строки в подчиненных таблицах.

DELETE может быть использован без фразы WHERE.

При этом удаляются **все строки** таблицы **без ограничений**.

Таблица, из которой удалены все строки, остается в базе данных и при желании может быть удалена из базы данных с помощью инструкции DROP TABLE.

Пример: *DELETE FROM Sales.SalesPerson;*

### **DELETE с использованием подзапроса**

DELETE может быть определен как подзапрос для того, чтобы удалить строки из базовой таблицы в зависимости от данных, хранящихся **в другой таблице**.

В данном примере удаляются строки из таблицы SalesPersonQuotaHistory на основе года до даты продажи, указанных в таблице SalesPerson.

```
DELETE FROM Sales.SalesPersonQuotaHistory
WHERE SalesPersonID IN
(SELECT SalesPersonID
FROM Sales.SalesPerson
WHERE SalesYTD > 2500000.00);
```

**Инструкция UPDATE** не изменяет количества строк в таблице, а только изменяет содержимое строк в указанных столбцах (или всей строки).

**Синтаксис команды:**

**UPDATE** *tab\_name*  
{**SET** *столбец\_1* = {значение | **DEFAULT** | **NULL**} [...n]  
[ **FROM** *имена\_таблиц условия\_выборки*] - **таблицы – источники**  
[**WHERE** *условие\_отбора\_строк*] - **строки, подлежащие обновлению**

Например, чтобы в таблице оценок (marks) прибавить всем студентам по одному баллу в виде бонуса можно записать выражение так:

**UPDATE marks SET mark=mark+1**

С помощью инструкции **UPDATE** данные можно модифицировать **только в одной таблице**.

При выполнении данная команда автоматически **проверяет все ограничения заданных столбцов**. Если хотя бы для одной изменяемой строки оно выполняется, **все действия**, уже выполненные командой будут отменены и дальнейшее ее выполнение прекращается.

В варианте обновления таблицы оценок из предыдущего примера, **в случае уже имеющейся хотя бы у одного студента высшей оценки 5 для всех остальных бонус будет не начислен**. Чтобы гарантированно выполнить это задание можно задать эту команду таким образом:

**UPDATE marks SET mark=mark+1 WHERE mark<5**

С помощью фразы **SET** можно присваивать не только конкретные значения, но также значения **NULL** и **DEFAULT**, если в данных столбцах обновляемой таблицы они допустимы или были определены при создании таблицы.

## **Задания для выполнения:**

### **Манипулирование данными при помощи DML – SQL**

**Использовать БД, созданную в ЛР-7.**

Все задания необходимо **выполнить средствами языка DML – SQL** (не через открытие и прямое редактирование таблиц посредством интерфейса Microsoft SQL Server Management Studio).

**Запросы обязательно сохранить** в отдельном файле (файлах) для повторной демонстрации преподавателю и распечатать в отчет.

После каждого шага **делать скриншоты в отчет**.

- ⇒ В таблицу **STUDENTS\_N** (**N – Ваш номер по журналу**) вставьте записи о 10-ти новых студентах (по своему усмотрению) со значением стипендии - 0.
- ⇒ В таблице **PREDMET\_N** **удалите ранее созданное в ЛР-8 ограничение** на названия предметов и вставьте новые записи о предметах «Базы данных», «Языки программирования», «Проектирование информационных систем» с

количеством часов, равных предметам «Физика», «Химия», «Философия» соответственно. В поле **TNUM** внесите значения NULL

- ⇒ В таблицу **PREDMET\_N** внесите изменения в поле **TNUM** для вставленных на предыдущем шаге записей, соответствующие тому, что преподаватель, который ведет предмет «Физика» будет вести и предмет «Базы данных», кто вел «Химия» - «Языки программирования» и соответственно «Философия» - «Проектирование информационных систем»
- ⇒ Внесите новые записи в таблицу **USP\_N** об оценках по всем новым предметам для всех студентов, имеющих стипендию 0 (оценки по собственному усмотрению)
- ⇒ Измените размер стипендии до 30 всем студентам, имеющим оценки «5» по более чем двум предметам (если таких нет – скорректируйте таблицы любым способом – через дополнительный запрос или через интерфейс Management Studio)
- ⇒ Измените размер стипендии до 5 всем студентам, имеющим хотя бы одну оценку «3» по предметам, при этом если у него стипендия не была равна «0»
- ⇒ Удалите записи из таблицы **STUDENTS\_N** о студентах, имеющих более 3-х оценок (2) по любым предметам. – **Обязательно сделайте** запрос на выборку всех строк таблицы и скриншот поместите в отчет предварительно **перед удалением и также после удаления.**