

Практическая работа № 10

Основы DML

1. Цель работы

1. Изучить команду INSERT.
2. Изучить команду UPDATE.
3. Изучить команду DELETE.
4. Изучить команду TRUNCATE TABLE.
5. Изучить конструкцию SELECT ... INTO.

2. Теоретическая часть

Команда INSERT осуществляет добавление данных в определенную таблицу. После команды INSERT можно добавить необязательное ключевое слово INTO. Упрощенный синтаксис команды имеет следующий вид:

```
INSERT INTO <таблица>
[(<список столбцов> ) ]
VALUES
(<список значений>)
```

Если добавляется две и более строки, тогда используется следующий синтаксис:

```
INSERT INTO <таблица>
[(<список столбцов> ) ]
VALUES
(<список значений>),
...
(<список значений>)
```

Количество и тип значений должны совпадать со списком столбцов. Последовательность столбцов может не совпадать с таблицей. Список столбцов должен быть заключен в круглые скобки, а его элементы должны разделяться запятыми.

Если столбец имеет свойства идентификатор, его нельзя указать в списке. Для таких столбцов сервер автоматически вычисляет новое значение.

Если столбец имеет свойство DEFAULT, при отсутствии его, в таблицу вставляется значение по умолчанию.

Если столбец имеет свойство NULL, при отсутствии его, в таблицу вставляется значение NULL.

Если столбец имеет свойство NOT NULL, его обязательно надо включить в список.

В списке значений для каждого столбца из указанных в списке столбцов должно быть одно значение. Список значений должен быть заключен в скобки.

Если значения в списке идут в том же порядке, как в таблице, и для каждого столбца таблицы определено значение, то список столбцов можно не указывать.

Если одновременно добавляется несколько строк значений, каждый список значений заключается в круглые скобки и разделяется запятыми.

Если значение для столбца неизвестно, и столбец имеет свойство NULL, для него в списке значений можно указать NULL (без кавычек).

Если требуется перенести строку из одной таблицы в другую таблицу можно использовать следующий синтаксис:

```
INSERT INTO <таблица>
[(<список столбцов>) ]
SELECT
    (<список столбцов>) ]
FROM
    исходная_таблица
WHERE
    <условие>
```

Типы данных в исходной и целевой таблицах должны совпадать.

Если таблицы имеют одинаковую структуру, можно после команды INSERT пропустить список столбцов, а после команды SELECT указать все столбцы, с помощью астериска «*».

Для изменения строк в таблице применяется команда UPDATE. Она имеет следующий формальный синтаксис:

```
UPDATE <таблица>
SET столбец1 = значение1, столбец2 = значение2, ..., столбецN = значениеN
[WHERE <условие>]
```

Использование условий необязательно, но тогда обновляются все строки таблицы. Рекомендуется сначала выполнять выборку строк с помощью SELECT, только потом использовать команду UPDATE.

Для удаления одной или нескольких строк из таблицы применяется команда DELETE. Она имеет следующий формальный синтаксис:

```
DELETE [FROM] <таблица>
[WHERE <условие>]
```

Ключевое слово FROM необязательно.

Использование условий необязательно, но тогда удаляются все строки таблицы. Рекомендуется сначала выполнять выборку строк с помощью SELECT, только потом использовать команду DELETE.

Для удаления всех строк из таблицы можно использовать команду TRUNCATE TABLE. Она имеет следующий формальный синтаксис:

```
TRUNCATE TABLE <таблица>
```

Инструкция TRUNCATE TABLE похожа на инструкцию DELETE без предложения WHERE, однако TRUNCATE TABLE выполняется быстрее и требует меньших ресурсов системы и журналов транзакций.

Для создания новой таблицы и ее заполнения можно использовать конструкцию SELECT...INTO. Она имеет следующий формальный синтаксис:

```
SELECT
<список столбцов>
INTO
<новая таблица>
FROM
<исходная таблица>
```

Столбцы в новой таблице создаются в порядке, соответствующем списку выбора и получают такие же имена, значения, типы данных и свойства допустимости значений NULL, которые указаны в соответствующем выражении в списке выбора.

3. Практическая часть

Таблица *Ученики*:

| ID | Фамилия | Предмет | Школа | Баллы |
|----|----------|------------|----------|-------|
| 1 | Иванова | Математика | Лицей | 98,5 |
| 2 | Петров | Физика | Лицей | 99 |
| 3 | Сидоров | Математика | Лицей | 88 |
| 4 | Полухина | Физика | Гимназия | 78 |
| 5 | Матвеева | Химия | Лицей | 92 |
| 6 | Касимов | Химия | Гимназия | 68 |
| 7 | Нурулин | Математика | Гимназия | 81 |
| 8 | Авдеев | Физика | Лицей | 87 |
| 9 | Никитина | Химия | Лицей | 94 |
| 10 | Барышева | Химия | Лицей | 88 |

Код для создания данной таблицы:

```
CREATE TABLE Ученики
(
    ID          INT PRIMARY KEY IDENTITY(1,1),
    Фамилия     VARCHAR(50) NOT NULL,
    Предмет     VARCHAR(50) NOT NULL,
    Школа       VARCHAR(50) NOT NULL,
    Баллы       FLOAT CHECK ((Баллы >= 0) AND (Баллы <= 100)) NULL
)
```

Пример 1: В таблицу «Ученики» внести новую запись для ученика гимназии Маркина, который по физике набрал 96 баллов:

```
INSERT INTO Ученики
    (Фамилия, Предмет, Школа, Баллы)
VALUES
    ('Маркин', 'Физика', 'Гимназия', 96)
```

Пример 2: В таблицу «Ученики» внести две строки, для ученицы лицея Никишиной, которая по химии набрала 77 баллов, и для ученика школы № 18 Андреева, оценка которого по математике неизвестна:

```
INSERT INTO Ученики
    (Фамилия, Предмет, Школа, Баллы)
VALUES
    ('Никишина', 'Химия', 'Лицей', 77),
    ('Андреев', 'Математика', 'Школа №18', NULL)
```

Пример 3: В таблице «Ученики» изменить данные Андреева, оценку исправить на 87:

```
UPDATE
    Ученики
SET
    Баллы = 87
WHERE
    Фамилия = 'Андреев'
```

Пример 4: В таблице «Ученики» изменить данные Никишиной, школу исправить на «Школа №31», а предмет на математику:

```
UPDATE
    Ученики
SET
    Школа = 'Школа №31', Предмет = 'Математика'
WHERE
    Фамилия = 'Никишина'
```

Пример 5: В таблице «Ученики» изменить данные всех учеников по математике, оценку уменьшить на 5 баллов:

```
UPDATE
    Ученики
SET
    Баллы = Баллы - 5
WHERE
    Предмет = 'Математика'
```

Пример 6: В таблице «Ученики» удалить данные всех учеников из школы №18:

DELETE FROM

Ученики

WHERE

Школа = 'Школа №18'

Пример 7: Создать таблицу «Лицейсты» и скопировать туда всех лицейстов:

SELECT

ID

,Фамилия

,Предмет

,Школа

,Баллы

INTO

Лицейсты

FROM

Ученики

WHERE

Школа = 'Лицей'

Пример 8: Очистить таблицу «Лицейсты»:

TRUNCATE TABLE Лицейсты

4. Задание

1. В таблицу «Ученики» внести новую запись для ученика школы № 18 Трошкова, оценка которого по химии неизвестна.
2. В таблицу «Ученики» внести три строки.
3. В таблице «Ученики» изменить данные Трошкова, школу исправить на № 21, предмет на математику, а оценку на 56.
4. В таблице «Ученики» изменить данные всех учеников по химии, оценку увеличить на 10%, если она ниже 60 баллов.
5. В таблице «Ученики» удалить данные всех учеников из школы №21.
6. Создать таблицу «Гимназисты» и скопировать туда данные всех гимназистов, кроме тех, которые набрали меньше 60 баллов.
7. Очистить таблицу «Гимназисты».

