

3. Манипулирование данными

3.1. Команды манипулирования данными

В SQL для выполнения операций ввода данных в таблицу, их изменения и удаления предназначены три команды языка манипулирования данными (DML). Это команды – **INSERT** (вставить), **UPDATE** (обновить), **DELETE** (удалить).

Команда **INSERT** осуществляет **вставку** в таблицу новой строки. В простейшем случае она имеет следующий вид:

```
INSERT INTO <имя таблицы> VALUES (<значение>, <значение>, ...);
```

При такой записи указанные в скобках после ключевого слова **VALUES** значения вводятся в поля добавленной в таблицу новой строки в том порядке, в котором соответствующие столбцы указаны при создании таблицы, то есть в операторе **CREATE TABLE**.

Например, ввод новой строки в таблицу **STUDENT** может быть осуществлен следующим образом

```
INSERT INTO STUDENT  
VALUES (101, 'Иванов', 'Александр', 200, 3, 'Москва', '6/10/1979', 15);
```

Чтобы такая команда могла быть выполнена, таблица с указанным в ней именем (**STUDENT**) должна быть предварительно определена (создана) командой **CREATE TABLE**. Если в какое-либо поле необходимо вставить **NULL**-значение, то оно вводится как обычное значение:

```
INSERT INTO STUDENT  
VALUES (101, 'Иванов', NULL, 200, 3, 'Москва', '6/10/1979', 15);
```

В случаях, когда необходимо ввести значения полей в порядке, отличном от порядка столбцов, заданного командой **CREATE TABLE**, или если требуется ввести значения не во все столбцы, то следует использовать следующую форму команды **INSERT**:

```
INSERT INTO STUDENT (STUDENT_ID, CITY, SURNAME, NAME)  
VALUES (101, 'Москва', 'Иванов', 'Саша');
```

Столбцам, наименования которых не указаны в приведенном в скобках списке, автоматически присваивается значение по умолчанию, если оно

назначено при описании таблицы (команда **CREATE TABLE**), либо значение **NULL**.

С помощью команды **INSERT** можно извлечь значение из одной таблицы и разместить его в другой, к примеру, запросом следующего вида:

```
INSERT INTO STUDENT1
SELECT *
FROM STUDENT
WHERE CITY = 'Москва';
```

При этом таблица STUDENT1 должна быть предварительно создана командой **CREATE TABLE** (раздел 4.1) и иметь структуру, идентичную таблице STUDENT.

Удаление строк из таблицы осуществляется с помощью команды **DELETE**.

Следующее выражение удаляет все строки таблицы EXAM_MARKS1.

```
DELETE FROM EXAM_MARKS1;
```

В результате таблица становится пустой (после этого она может быть удалена командой **DROP TABLE**).

Для удаления из таблицы сразу нескольких строк, удовлетворяющих некоторому условию можно воспользоваться предложением **WHERE**, например,

```
DELETE FROM EXAM_MARKS1
WHERE STUDENT_ID = 103;
```

Можно удалить группу строк

```
DELETE FROM STUDENT1
WHERE CITY = 'Москва';
```

Команда **UPDATE** позволяет **изменять**, то есть обновлять, значения некоторых или всех полей в существующей строке или строках таблицы. Например, чтобы для всех университетов, сведения о которых находятся в таблице UNIVERSITY1, изменить рейтинг на значение 200, можно использовать конструкцию:

```
UPDATE UNIVERSITY1
SET RATING = 200;
```

Для указания конкретных строк таблицы, значения полей которых должны быть изменены, в команде **UPDATE** можно использовать предикат, указываемый в предложении **WHERE**.

```
UPDATE UNIVERSITY1
SET RATING = 200
WHERE CITY = 'Москва';
```

В результате выполнения этого запроса будет изменен рейтинг только у университетов, расположенных в Москве.

Команда **UPDATE** позволяет изменять не только один, но и множество столбцов. Для указания конкретных столбцов, значения которых должны быть модифицированы, используется предложение **SET**.

Например, наименование предмета обучения 'Математика' (для него SUBJ_ID = 43) должно быть заменено, на название 'Высшая математика', при этом идентификационный номер необходимо сохранить, но в соответствующие поля строки таблицы ввести новые данные об этом предмете обучения. Запрос будет выглядеть следующим образом.

```
UPDATE SUBJECT1
SET SUBJ_NAME = 'Высшая математика', HOUR = 36, SEMESTER
= 1
WHERE SUBJ_ID = 43;
```

В предложении **SET** команды **UPDATE** можно использовать скалярные выражения, указывающие способ изменения значений поля, в которые могут входить значения изменяемого и других полей.

```
UPDATE UNIVERSITY1
SET RATING = RATING*2;
```

Например, для увеличения в таблице STUDENT1 значения поля STIPEND в два раза для студентов из Москвы можно использовать запрос

```
UPDATE STUDENT1
  SET STIPEND = STIPEND*2
  WHERE CITY = 'Москва';
```

Предложение **SET** не является предикатом, поэтому в нем можно указать значение **NULL** следующим образом.

```
UPDATE UNIVERSITY1
  SET RATING = NULL
  WHERE CITY = 'Москва';
```

УПРАЖНЕНИЯ

86.Напишите команду, которая вводит в таблицу SUBJECT строку для нового предмета обучения со следующими значениями полей:

```
SEMESTER = 4; SUBJ_NAME = 'Алгебра'; HOUR = 72;
SUBJ_ID = 201.
```

87.Введите запись для нового студента, которого зовут Орлов Николай, обучающегося на первом курсе ВГУ, живущего в Воронеже, сведения о дате рождения и размере стипендии не известны.

88.Напишите команду, удаляющую из таблицы EXAM_MARKS записи обо всех оценках студента, идентификатор которого равен 100.

89.Напишите команду, которая увеличивает на 5 значение рейтинга всех, имеющих в базе данных университетов, расположенных в Санкт-Петербурге.

90.Измените в таблице значение города, в котором проживает студент Иванов, на “Воронеж”.