Тема 22

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 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', 2000, 3,'Москва’, '6/10/2004', 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, 2000, 3, 'Москва','6/10/2004', 15);

В случаях, когда необходимо ввести значения полей в порядке, отличном от порядка столбцов, заданного командой CREATE TABLE, или требуется ввести значения не во все столбцы, 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, например, запросом следующего вида:

INSERT INTO STUDENT1

SELECT \*

FROM STUDENT

WHERE CITY = 'Москва';

При этом таблица STUDENTI должна быть предварительно создана командой CREATE TABLE 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:

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 UNIVERSITYI, 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 STUDENT 1

SET STIPEND = STIPEND\*2

WHERE CITY = 'Москва';

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

UPDATE UNIVERSITYl

SET RATING = NULL

WHERE CITY = 'Москва';

Применение оператора INSERT с подзапросом позволяет загружать сразу несколько строк в одну таблицу, используя информацию из другой таблицы.

В то время как оператор INSERT, использующий VALUES, добавляет только одну строку, INSERT с подзапросом добавляет в таблицу столько строк, сколько подзапрос извлекает из другой таблицы.

При этом количество и тип возвращаемых подзапросом столбцов должен соответствовать количеству и типу столбцов таблицы, в которую вставляются данные.

Например, пусть таблица STUDENT1 имеет структуру, полностью совпадающую со структурой таблицы STUDENT.

Запрос, позволяющий заполнить таблицу STUDENT1 записями обо всех

студентах из Москвы из таблицы STUDENT, выглядит следующим образом:

INSERT INTO STUDENT1

SELECT \*

FROM STUDENT

WHERE CITY = 'Москва';

Чтобы добавить в таблицу STUDENT1 сведения обо всех студентах, которые учатся в Москве, можно использовать в предложении WHERE соответствующий подзапрос.

INSERT INTO STUDENT1

SELECT \*

FROM STUDENT

WHERE UNIV\_ID IN

(SELECT UNIV\_ID

FROM UNIVERSITY

WHERE CITY = 'Москва');

Предположим, существует таблица SSTUD, в которой хранятся сведения о студентах, обучающихся в том же городе, в котором они живут. Можно заполнить эту таблицу данными из таблицы STUDENT, используя связанные подзапросы, следующим образом:

INSERT INTO SSTUD

SELECT \*

FROM STUDENT A

WHERE CITY IN

(SELECT CITY

FROM UNIVERSITY В

WHERE A.UNIV\_ID = B.UNIV\_ID);

Предположим, требуется выбрать список студентов, имеющих максимальный балл на каждый день сдачи экзаменов, и разместить его в другой таблице с именем EXAM. Это можно осуществить с помощью запроса

INSERT INTO EXAM

SELECT EXAM\_ID, STUDENT\_ID, SUBJ\_ID, MARK, EXAM\_DATE

FROM EXAM\_MARKS A

WHERE MARK =

(SELECT MAX (MARK)

FROM EXAM\_MARKS В

WHERE A.EXAM\_DATE = В.EXAM\_DATE);

*Использование подзапросов с DELETE*

Пусть филиал университета в Нью-Васюках ликвидирован и требуется удалить из таблицы STUDENT записи о студентах, которые там учились. Эту операцию можно выполнить с помощью запроса

DELETE

FROM STUDENT

WHERE UNIV\_ID IN

(SELECT UNIV\_ID

FROM UNIVERSITY

WHERE CITY = 'Нью-Васюки');

В предикате предложения FROM (подзапроса) нельзя ссылаться на таблицу, из которой осуществляется удаление. Однако можно ссылаться на текущую строку из таблицы, являющуюся кандидатом на удаление, то есть на строку, которая в настоящее время проверяется в основном предикате.

DELETE

FROM STUDENT

WHERE EXISTS

(SELECT \*

FROM UNIVERSITY

WHERE RATING = 401

AND STUDENT.UNIV\_ID = UNIVERSITY. UNIV\_ID);

Часть AND предиката внутреннего запроса ссылается на таблицу STUDENT. Команда удаляет данные о студентах, которые учатся в университетах, имеющих рейтинг, равный 401.

Существуют и другие способы решения этой задачи.

DELETE

FROM STUDENT

WHERE 401 IN

(SELECT RATING

FROM UNIVERSITY

WHERE STUDENT. UNIV\_ID = UNIVERSITY. UNIV\_\_ID);

Пусть нужно найти наименьшее значение оценки, полученной в каждый день сдачи экзаменов, и удалить из таблицы сведения о студенте, который получил эту оценку. Запрос будет иметь вид:

DELETE

FROM STUDENT

WHERE STUDENT\_ID IN

(SELECT STUDENT\_ID

FROM EXAM\_MARKS A

WHERE MARK=

(SELECT MIN (MARK)

FROM EXAM\_MARKS В

WHERE A.EXAM\_DATE = В.EXAMJDATE));

Так как столбец STUDENT\_ID является первичным ключом,

то удаляется единственная строка.

Если в какой-то день сдавался только один экзамен (то есть получена только одна минимальная оценка) и по какой-либо причине запись, в которой находится эта оценка, требуется оставить, то решение будет иметь вид:

DELETE

FROM STUDENT

WHERE STUDENT\_ID IN

(SELECT STUDENT\_ID

FROM EXAM\_MARKS A

WHERE MARK =

(SELECT MIN (MARK)

FROM EXAM\_MARKS В

WHERE A.EXAM\_DATE = B.EXAM\_DATE

AND 1 <

(SELECT COUNT (SUBJ\_ID)

FROM EXAM\_MARKS В

WHERE A.EXAM\_DATE = В.EXAM\_DATE)));

С помощью команды UPDATE можно применять подзапросы в любой форме, приемлемой для команды DELETE.

Например, используя связанные подзапросы, можно увеличить значение размера стипендии на 2000 в записях студентов, сдавших экзамены на 4 и 5.

UPDATE STUDENT1

SET STIPEND = STIPEND + 2000

WHERE 4 <=

(SELECT MIN (MARK)

FROM EXAM\_MARKS

WHERE EXAM\_MARKS.STUDENT\_ID = STUDENT1.STUDENT\_ID);

Другой запрос: «Уменьшить величину стипендии на 20 всем

студентам, получившим на экзамене минимальную оценку».

UPDATE STUDENT1

SET STIPEND = STIPEND - 20

WHERE STUDENT\_ID IN

(SELECT STUDENT\_ID

FROM EXAM\_MARKS A

WHERE MARK =

(SELECT MIN (MARK)

FROM EXAM\_MARKS В

WHERE A.EXAM DATE = B.EXAM DATE));