

СРСП 10. Предложения модификации данных

Модификация данных может выполняться с помощью предложений DELETE (удалить), INSERT (вставить) и UPDATE (обновить). Они могут оперировать как базовыми таблицами, так и представлениями. Однако по ряду причин не все представления являются обновляемыми.

Предложение DELETE имеет формат

DELETE

FROM базовая таблица | представление
[WHERE фраза];

и позволяет удалить содержимое всех строк указанной таблицы (при отсутствии WHERE фразы) или тех ее строк, которые выделяются WHERE фразой.

Удаление единственной записи

Удалить поставщика с ПС = 7.

DELETE

FROM Поставщики
WHERE ПС = 7;

Если таблица Поставки содержит в момент выполнения этого предложения какие-либо поставки для поставщика с ПС = 7, то такое удаление нарушит непротиворечивость базы данных. К сожалению нет операции удаления, одновременно воздействующей на несколько таблиц.

Удаление множества записей

Удалить все поставки.

DELETE

FROM Поставки;

Поставки - все еще известная таблица, но в ней теперь нет строк. Для уничтожения таблицы надо выполнить операцию DROP TABLE Поставки.

Удалить все мясные блюда.

DELETE FROM Блюда

WHERE Основа = 'Мясо';

Удаление с вложенным подзапросом

Удалить все поставки для поставщика из Паневежиса.

DELETE

FROM Поставки

WHERE ПС IN

(SELECT ПС

FROM Поставщики

WHERE Город = 'Паневежис');

Предложение INSERT имеет один из следующих форматов:

INSERT

INTO {базовая таблица | представление} [(столбец [,столбец] ...)]

VALUES ({константа | переменная} [, {константа | переменная}] ...);

или

INSERT

INTO {базовая таблица | представление} [(столбец [,столбец] ...)]

подзапрос;

В первом формате в таблицу вставляется строка со значениями полей, указанными в перечне фразы VALUES (значения), причем i-е значение соответствует i-му столбцу в списке столбцов (столбцы, не указанные в списке, заполняются NULL-значениями).

Если в списке VALUES фразы указаны все столбцы модифицируемой таблицы и порядок их перечисления соответствует порядку столбцов в описании таблицы, то список столбцов в фразе INTO можно опустить. Однако не советуем этого делать, так как при изменении описания таблицы (перестановка столбцов или изменение их числа) придется переписывать и INSERT предложение.

Во втором формате сначала выполняется подзапрос, т.е. по предложению SELECT в памяти формируется рабочая таблица, а потом строки рабочей таблицы загружаются в модифицируемую таблицу. При этом i-й столбец рабочей таблицы (i-й элемент списка SELECT) соответствует i-му столбцу в списке столбцов модифицируемой таблицы. Здесь также при выполнении указанных выше условий может быть опущен список столбцов фразы INTO.

Вставка единственной записи в таблицу

Добавить в таблицу Блюда блюдо:

Шашлык (БЛ - 34, Блюдо - Шашлык, В - Г, Основа - Мясо, Выход - 150)

при неизвестной пока трудоемкости приготовления этого блюда.

INSERT

INTO Блюда (БЛ, Блюдо, В, Основа, Выход)

VALUES (34, 'Шашлык', 'Г', 'Мясо', 150);

Создается новая запись для блюда с номером 34, с неопределенным значением в столбце Труд.

Порядок полей в INSERT не обязательно должен совпадать с порядком полей, в котором они определялись при создании таблицы. Вполне допустима и такая версия предыдущего предложения:

INSERT

INTO Блюда (Основа, В, Блюдо, БЛ, Выход)

VALUES ('Мясо', 'Г', 'Шашлык', 34, 150);

При известной трудоемкости приготовления шашлыка (например, 5 коп) сведения о нем можно ввести с помощью укороченного предложения:

INSERT

INTO Блюда

VALUES (34, 'Шашлык', 'Г', 'Мясо', 150, 5);

в котором должен соблюдаться строгий порядок перечисления вводимых значений, так как, не имея перечня загружаемых столбцов, СУБД может использовать лишь перечень, который определен при создании модифицируемой таблицы.

Вставка множества записей

Создать временную таблицу К_меню, содержащую калорийность и стоимость всех блюд, которые можно приготовить из имеющихся продуктов.

Для создания описания временной таблицы можно, например, воспользоваться предложением CREATE TABLE (см.п.5.2)

CREATE TABLE К_меню

(Вид CHAR (10),
Блюдо CHAR (60),
Калор_блюда INTEGER,
Стоим_блюда REAL);

а для ее загрузки данными – предложение INSERT с вложенным подзапросами:

INSERT

INTO К_меню

SELECT Вид, Блюдо,

INT(SUM(((Белки+Углев)*4.1+Жиры*9.3) * Вес/1000)),

(SUM(Стоимость/К_во*Вес/1000) + MIN(Труд/100))

```
FROM Блюда, Вид_блюд, Состав, Продукты, Наличие
WHERE Блюда.БЛ = Состав.БЛ
AND Состав.ПР = Продукты.ПР
AND Состав.ПР = Наличие.ПР
AND Блюда.В = Вид_блюд.В
AND БЛ NOT IN
  (
    SELECT БЛ
    FROM Состав
    WHERE ПР IN
      (
        SELECT ПР
        FROM Наличие
        WHERE К_во = 0))
GROUP BY Вид, Блюдо
ORDER BY Вид, 3;
```

В этом запросе предложение `SELECT` выполняется так же, как обычно), но результат не выводится на экран, а копируется в таблицу `К_меню`. Теперь с этой копией можно работать как с обычной базовой таблицей (Блюда, Продукты,...), т.е. выбирать из нее данные на экран или принтер, обновлять в ней данные и т.п. Никакая из этих операций не будет оказывать влияния на исходные данные.

Предложение `UPDATE` также имеет два формата. Первый из них:

```
UPDATE {базовая таблица | представление}
SET   столбец = значение [, столбец = значение] ...
[WHERE фраза]
где значение - это
```

столбец | выражение | константа | переменная

и может включать столбцы лишь из обновляемой таблицы, т.е. значение одного из столбцов модифицируемой таблицы может заменяться на значение ее другого столбца или выражения, содержащего значения нескольких ее столбцов, включая изменяемый.

При отсутствии `WHERE` фразы обновляются значения указанных столбцов во всех строках модифицируемой таблицы. `WHERE` фраза позволяет сократить число обновляемых строк, указывая условия их отбора.

Второй формат описывает предложение, позволяющее производить обновление значений модифицируемой таблицы по значениям столбцов из других таблиц. К сожалению в ряде СУБД эти форматы отличаются друг от друга и от стандарта. Для примера приведем один из таких форматов:

```
UPDATE {базовая таблица | представление}
SET   столбец = значение [, столбец = значение] ...
FROM  {базовая таблица | представление} [псевдоним],
      {базовая таблица | представление} [псевдоним]
      [, {базовая таблица | представление} [псевдоним]] ...
[WHERE фраза]
```

Здесь перечень таблиц фразы `FROM` содержит имя модифицируемой таблицы и тех таблиц, значения столбцов которых используются для обновления. При этом, естественно, таблицы должны быть связаны между собой в `WHERE` фразе, которая, кроме того, служит для указания условий отбора обновляемых строк модифицируемой таблицы.

В значениях, находящихся в правых частях равенств фразы `SET`, следует уточнять имена используемых столбцов, предваряя их именем таблицы (псевдонима).

Обновление единственной записи

Изменить название блюда с кодом БЛ=5 на Форшмак, увеличить его выход на 30 г и установить NULL-значение в столбец Труд.

UPDATE Блюда

```
SET    Блюдо = 'Форшмак', Выход = (Выход+30), Труд = NULL  
WHERE БЛ = 5;
```

Обновление множества записей

Утроить цену всех продуктов таблицы поставки (кроме цены кофе - ПР = 17).

UPDATE Поставки

```
SET    Цена = Цена * 3  
WHERE ПР <> 17;
```

Обновление с подзапросом

Установить равной нулю цену и К_во продуктов для поставщиков из Паневежиса и Резекне.

UPDATE Поставки

```
SET    Цена = 0, К_во = 0
```

WHERE ПС IN

```
(SELECT ПС  
      FROM    Поставщики  
     WHERE Город IN ('Паневежис', 'Резекне'));
```

Обновление нескольких таблиц

Изменить номер продукта ПР = 13 на ПР = 20.

UPDATE Продукты UPDATEСостав

```
SET    ПР = 20      SET    ПР = 20  
WHERE ПР = 13;    WHERE ПР = 13;
```

UPDATE Поставки UPDATEНаличие

```
SET    ПР = 20      SET    ПР = 20  
WHERE ПР = 13;    WHERE ПР = 13;
```

К сожалению, в единственным запросе невозможно обновить более одной таблицы, а так как код продукта входит в четыре таблицы, то пришлось выдать четыре сходных запроса. Это может привести к противоречию базы данных (нарушению целостности по ссылкам), поскольку после выполнения первого предложения таблицы Состав, Поставки и Наличие ссылаются на уже несуществующий продукт. База становится непротиворечивой только после выполнения четвертого запроса.

Задание к СРСП 10:

Исходные данные: Пусть имеются таблицы БД со следующей структурой

Продавцы (Таб.номер, ФИО, город)

Товары (Номер товара, Наименование)

Движение товаров (дата, таб.номер продавца, код товара, цена, ФИО покупателя)

1. Напишите команду которая бы поместила следующие значения, в их нижеуказанном порядке, в таблицу Продавцов:

город - Караганда,

ФИО - Иванов,

Таб.номер - 1100.

2. Напишите команду которая бы удалила все покупки покупателя Петрова из таблицы Движение товаров.

3. Напишите команду которая бы увеличила цену всех товаров в таблице Движение товаров на 100.

4. Продавец Иванов оставил компанию. Переназначьте его заказчиков продавцу Сидорову.