МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Череповецкий государственный университет»

**Лабораторная работа № 3**

**«SQL. Язык манипулирования данными (DML)»**

**Выполнил:**

студент гр. 1ИВТпб-01-31оп

Климов А.Г.  
**Проверил:**  преподаватель

Селяничев О.Л.  
Отметка о зачете:

Череповец

2017 год

**Задания**

SQL. Язык манипулирования данными (DML) продолжение.

1. Работайте с ранее созданной БД ПАНСИОН.
2. Ознакомьтесь с теорией в главах 4 и 5 (список необходимых для изучения пунктов – ниже). Каждый встретившийся запрос проверьте практически -воспроизведите запрос в программе DbLoader. Изменяйте всякий запрос по-своему так, чтобы и он давал правильное решение (естественно, другое).

Пункт 4.2. на практике не выполнять – лишь изучить теоретически.

1. В отчет по ЛР должны попасть задания и ответы на них – из учебника и свое.

Глава 4. Предложения модификации данных SQL

4.1. Особенности и синтаксис предложений модификации

4.2. Предложение DELETE

4.2.1. Удаление единственной записи

4.2.2. Удаление множества записей

4.2.3. Удаление с вложенным подзапросом

4.3. Предложение INSERT

4.3.1. Вставка единственной записи в таблицу

4.3.2. Вставка множества записей

4.3.3. Использование INSERT...SELECT для построения внешнего соединения

4.4. Предложение UPDATE

4.4.1. Обновление единственной записи

4.4.2. Обновление множества записей

4.4.3. Обновление с подзапросом

4.4.4. Обновление нескольких таблиц

4.5. О конструировании предложений модификации

Глава 5. О предложениях определения данных и оптимизации запросов

5.1. Системный каталог

5.2. Создание и уничтожение базовых таблиц

5.3. О индексах и производительности

5.4. Представления

5.4.1. Создание и уничтожение представлений

5.4.2. Операции выборки из представлений

5.4.3. Обновление представлений

5.4.4. Для чего нужны представления

**Выполнение заданий**

**Глава 4. Предложения модификации данных SQL**

**4.1. Особенности и синтаксис предложений модификации**

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

**4.2. Предложение DELETE**

**4.2.1. Удаление единственной записи**

Таблица 1

|  |  |
| --- | --- |
| Из учебника | Мое |
| Удалить поставщика с ПС = 7 | Удалить поставщика с ПС = 1 и ПС = 7 |
| DELETE  FROM Поставщики  WHERE ПС = 7; | DELETE  FROM Поставщики  WHERE ПС = 1 OR ПС = 7; |

**4.2.2. Удаление множества записей**

Таблица 2

|  |  |
| --- | --- |
| Из учебника | Мое |
| Удалить все поставки | Удалить всех поставщиков |
| DELETE  FROM Поставки; | DELETE  FROM Поставщики; |

Таблица 3

|  |  |
| --- | --- |
| Из учебника | Мое |
| Удалить все мясные блюда | Удалить все мясные блюда и рыбные блюда |
| DELETE FROM Блюда  WHERE Основа = 'Мясо'; | DELETE FROM Блюда  WHERE Основа = 'Мясо' OR Основа = 'Рыба'; |

**4.2.3. Удаление с вложенным подзапросом**

Таблица 4

|  |  |
| --- | --- |
| Из учебника | Мое |
| Удалить все поставки для поставщика из Паневежиса | Удалить все поставки с судаком или в которых не содержится белка |
| DELETE  FROM Поставки  WHERE ПС IN  (SELECT ПС  FROM Поставщики  WHERE Город = 'Паневежис'); | DELETE  FROM Поставки  WHERE ПР IN  (SELECT ПР  FROM Продукты  WHERE Продукт = 'Судак' OR Белки = 0); |

**4.3. Предложение INSERT**

**4.3.1. Вставка единственной записи в таблицу**

Таблица 5

|  |  |
| --- | --- |
| Из учебника | Мое |
| Добавить в таблицу Блюда блюдо при неизвестной пока трудоемкости его приготовления | Добавить в таблицу Блюда блюдо при неизвестной пока трудоемкости его приготовления |
| INSERT  INTO Блюда (БЛ, Блюдо, В, Основа, Выход)  VALUES (34, 'Шашлык', 'Г', 'Мясо', 150); | INSERT INTO Блюда(БЛ, Блюдо, В, Основа, Выход) VALUES (35, 'Курица в кляре', 'Г', 'Мясо', 200); |

Таблица 6

|  |  |
| --- | --- |
| Из учебника | Мое |
| Добавить в таблицу Блюда блюдо при неизвестной пока трудоемкости его приготовления. Порядок полей в INSERT не обязательно должен совпадать | Добавить в таблицу Блюда блюдо при неизвестной пока трудоемкости его приготовления. Порядок полей в INSERT не обязательно должен совпадать |
| INSERT  INTO Блюда (Основа, В, Блюдо, БЛ, Выход)  VALUES ('Мясо', 'Г', 'Шашлык', 34, 150); | INSERT INTO Блюда(Основа, В, Блюдо, БЛ, Выход) VALUES ('Овощи', ‘Г’, ‘Салат зимний’, 36, 120); |

Таблица 7

|  |  |
| --- | --- |
| Из учебника | Мое |
| При известной трудоемкости приготовления шашлыка (например, 5 коп) ввести сведения о нем | При известной трудоемкости приготовления шашлыка (например, 15 коп) ввести сведения о нем |
| INSERT  INTO Блюда  VALUES (34, 'Шашлык', 'Г', 'Мясо', 150, 5); | INSERT INTO Блюда VALUES (35, 'Курица в кляре', 'Г', 'Мясо', 200, 15); |

Таблица 8

|  |  |
| --- | --- |
| Из учебника | Мое |
| Добавить в рецепт блюда Салат летний (БЛ = 1) немного (15 г) лука (ПР = 10) | Добавить в рецепт блюда Курица в кляре (БЛ = 35) немного (25 мл) масла (ПР = 3) |
| INSERT  INTO Состав (БЛ, ПР, Вес)  VALUES (1, 10, 15); | INSERT  INTO Состав (БЛ, ПР, Вес)  VALUES (35, 3, 25); |

**4.3.2. Вставка множества записей**

Таблица 9

|  |  |
| --- | --- |
| Из учебника | Мое |
| Создать временную таблицу К\_меню, содержащую калорийность и стоимость всех блюд, которые можно приготовить из имеющихся продуктов. Эта таблица будет использоваться шеф-поваром для составления меню на следующий день. | Создать временную таблицу К\_труд, содержащую Блюда и труд, для целой недели |
| CREATE TABLE К\_меню  ( Вид CHAR (10),  Блюдо CHAR (60),  Калор\_блюда INTEGER,  Стоим\_блюда REAL); | CREATE TABLE К\_труд  (Блюдо CHAR (20),  Труд int); |

**4.4. Предложение UPDATE**

**4.4.1. Обновление единственной записи**

Таблица 10

|  |  |
| --- | --- |
| Из учебника | Мое |
| Изменить название блюда с кодом БЛ=5 на Форшмак, увеличить его выход на 30 г и установить NULL-значение в столбец Труд | Изменить название блюда с кодом БЛ=34 на Форшмак, увеличить его выход на 52 г и установить NULL-значение в столбец Труд |
| UPDATE Блюда  SET Блюдо = 'Форшмак', Выход = (Выход+30), Труд = NULL  WHERE БЛ = 5; | UPDATE Блюда  SET Блюдо = 'Стейк из говядины', Выход = (Выход+52), Труд = NULL  WHERE БЛ = 34; |

**4.4.2. Обновление множества записей**

Таблица 11

|  |  |
| --- | --- |
| Из учебника | Мое |
| Утроить цену всех продуктов таблицы поставки (кроме цены кофе - ПР = 17) | Увеличить цену всех продуктов таблицы поставки (кроме цены говядины - ПР = 1 и лука – ПР = 10) |
| UPDATE Поставки  SET Цена = Цена \* 3  WHERE ПР <> 17; | UPDATE Поставки  SET Цена = Цена + 5  WHERE ПР <> 1 OR ПР <> 10; |

**4.4.3. Обновление с подзапросом**

Таблица 12

|  |  |
| --- | --- |
| Из учебника | Мое |
| Установить равной нулю цену и К\_во продуктов для поставщиков из Паневежиса и Резекне | Установить равной нулю цену и К\_во продуктов для яйца и сметаны |
| UPDATE Поставки  SET Цена = 0, К\_во = 0  WHERE ПС IN  (SELECT ПС  FROM Поставщики  WHERE Город IN ('Паневежис', 'Резекне')); | UPDATE Поставки  SET Цена = 0, К\_во = 0  WHERE ПР IN  (SELECT ПР  FROM Продукты  WHERE Продукт IN ('Яйца', 'Сметана')); |

**4.5. О конструировании предложений модификации**

Создаём таблицу “ Выбор”:

**CREATE TABLE Выбор(СМ int, Т int, В char(1), БЛ int);**

Заносим записи в созданную таблицу Рис. 1:

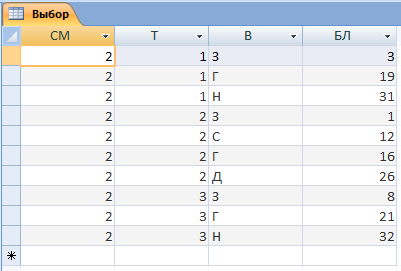


Рис. 1. Записи таблицы “ Выбор”

Создаём таблицу “ Выбрано”:

**CREATE TABLE Выбрано(СМ int, Т int, БЛ int);**

Заносим записи в созданную таблицу Рис. 2:

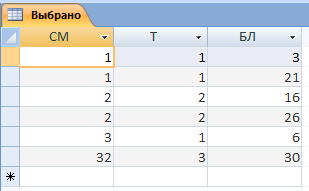


Рис. 2. Записи таблицы “ Выбрано”

Таблица 13

|  |  |
| --- | --- |
| Из учебника | Мое |
| Если в WHERE фразе предложений DELETE и UPDATE используется вложенный подзапрос, то во фразе FROM этого подзапроса не должна упоминаться таблица, из которой удаляются (в которой обновляются) строки. Аналогично, в подзапросе предложения INSERT не должна упоминаться таблица, в которую загружаются данные.  Так, SQL отвергнет предложение |  |
| INSERT  INTO Выбрано  SELECT (33), Т, БЛ  FROM Выбрано  WHERE СМ = 17; |  |

Таблица 14

|  |  |
| --- | --- |
| Из учебника | Мое |
| Ввести информацию о том, что отдыхающий, сидящий на 33-м месте, выбирает тот же набор блюд, что и отдыхающий, сидящий на 17-м месте. Ввод осуществить через какую-либо промежуточную таблицу, например, таблицу Выбор |  |
| DELETE  FROM Выбор;  INSERT  INTO Выбор (СМ, Т, БЛ)  SELECT (33), Т, БЛ  FROM Выбрано  WHERE СМ = 17;  INSERT  INTO Выбрано  SELECT СМ, Т, БЛ  FROM Выбор; |  |

**Глава 5. О предложениях определения данных и оптимизации запросов**

**5.3. О индексах и производительности**

Таблица 15

|  |  |
| --- | --- |
| Из учебника | Мое |
| Создать индексы для столбцов БЛ и Основа таблицы Блюда |  |
| CREATE UNIQUE INDEX Блюда\_БЛ ON Блюда (БЛ);  CREATE INDEX Блюда\_Основа ON Блюда (Основа); |  |

Таблица 16

|  |  |
| --- | --- |
| Из учебника | Мое |
| Создать индекс для первичного ключа (столбцы БЛ и ПР) таблицы Состав |  |
| CREATE UNIQUE INDEX Состав\_БЛ\_ПР ON Состав (БЛ, ПР); |  |

**5.4. Представления**

**5.4.1. Создание и уничтожение представлений**

Таблица 17

|  |  |
| --- | --- |
| Из учебника | Мое |
| Создать представление Мясные\_блюда |  |
| CREATE VIEW Мясные\_блюда  AS SELECT БЛ, Блюдо, В, Выход  FROM Блюда  WHERE Основа = 'Мясо'; | CREATE VIEW Молочные\_блюда  AS SELECT БЛ, Блюдо, В, Выход  FROM Блюда  WHERE Основа = 'Молоко'; |

**5.4.2. Операции выборки из представлений**

Таблица 18

|  |  |
| --- | --- |
| Из учебника | Мое |
| Осуществить выборку из базовых таблиц и из представлений |  |
| CREATE VIEW Горячие\_мясные\_блюда  AS SELECT Блюдо, Продукт, Вес  FROM Мясные\_блюда, Состав, Продукты  WHERE Мясные\_блюда.БЛ = Состав.БЛ  AND Продукты.ПР = Состав.ПР  AND В = 'Г'; | CREATE VIEW З\_молочные\_блюда  AS SELECT Блюдо, Продукт, Вес  FROM Молочные\_блюда, Состав, Продукты  WHERE Молочные\_блюда.БЛ = Состав.БЛ  AND Продукты.ПР = Состав.ПР  AND В = 'З'; |

Таблица 19

|  |  |
| --- | --- |
| Из учебника | Мое |
| Запрос на получение всех данных из представления | Запрос на получение всех данных из представления |
| SELECT \*  FROM Мясные\_блюда; | SELECT \*  FROM Молочные\_блюда; |

Таблица 20

|  |  |
| --- | --- |
| Из учебника | Мое |
| Получить сведения о горячих мясных блюдах, в состав которых входят помидоры | Получить сведения о молочных блюдах, в состав которых входит крупа |
| SELECT Блюдо, Продукт, Вес  FROM Горячие\_мясные\_блюда  WHERE Блюдо IN  ( SELECT Блюдо  FROM Горячие\_мясные\_блюда  WHERE Продукт = 'Помидоры') | SELECT Блюдо, Продукт, Вес  FROM З\_молочные\_блюда  WHERE Блюдо IN  ( SELECT Блюдо  FROM З\_молочные\_блюда  WHERE Продукт = 'Крупа') |

**5.4.3. Обновление представлений**

Безусловно обновляемыми являются представления, полученные из единственной базовой таблицы простым исключением некоторых ее строк и (или) столбцов, обычно называемые "представление-подмножество строк и столбцов". Таким является представление Мясные\_блюда, полученное из базовой таблицы Блюда исключением из нее столбца Труд и строк, не содержащих значение 'Мясо' в столбце Основа. Работая с ним, можно:

вставить (операция INSERT) новую строку, например, строку (34, 'Шашлык', 'Г', 150), фактически вставляя соответствующую строку (34, 'Шашлык', 'Г', 150, NULL) в лежащую в основе базовую таблицу Блюда;

удалить (операция DELETE) существующую строку из представления, например строку (13, 'Бастурма', 'Г', 300), фактически удаляя соответствующую строку (13, 'Бастурма', 'Г', 300, 5) из таблицы Блюда;

обновить (операция UPDATE) какое-либо поле в существующей строке, например увеличить массу порции Бефстроганова с 210 до 250 граммов, фактически осуществляя то же самое изменение в соответствующем поле таблицы Блюда.

**5.4.4. Для чего нужны представления**

Одна из основных задач, которую позволяют решать представления, - обеспечение независимости пользовательских программ от изменения логической структуры базы данных при ее расширении и (или) изменении размещения столбцов, возникающего, например, при расщеплении таблиц. В последнем случае можно создать представление-соединение с именем и структурой расщепленной таблицы, позволяющее сохранить программы, существовавшие до изменения структуры базы данных.

Кроме того, представления дают возможность различным пользователям по-разному видеть одни и те же данные, возможно, даже в одно и то же время. Это особенно ценно при работе различных категорий пользователей с единой интегрированной базой данных. Пользователям предоставляют только интересующие их данные в наиболее удобной для них форме (окно в таблицу или в любое соединение любых таблиц).

Наконец, от определенных пользователей могут быть скрыты некоторые данные, невидимые через предложенное им представление. Таким образом, принуждение пользователя осуществлять доступ к базе данных через представления является простым, но эффективным механизмом для управления санкционированием доступа.