Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт о лабораторной работе №4**

**Дисциплина:** Базы данных

**Тема:** Язык SQL-DML

Выполнил студент гр. 43501/1 Калугина М.О.

Руководитель Мяснов А.В.

“ ” 2016 г.

Санкт -Петербург

2016

1. **Цели работы**

Познакомить студентов с языком создания запросов управления данными SQL-DML.

# Программа работы

* Изучите SQL-DML
* Выполните все запросы из списка стандартных запросов. Продемонстрируйте результаты преподавателю.
* Получите у преподавателя и реализуйте SQL-запросы в соответствии с индивидуальным заданием. Продемонстрируйте результаты
* Выполненные запросы SELECT сохраните в БД в виде представлений, запросы INSERT, UPDATE или DELETE -- в виде ХП. Выложите скрипт в Subversion.

1. **Ход работы**

DML — это семейство компьютерных языков, используемых в компьютерных программах или пользователями [баз данных д](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)ля получения, вставки, удаления или изменения данных в [базах данных.](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)

Функции языков DML определяются первым словом в предложении (часто называемом запросом), которое почти всегда является глаголом. В случае с SQL эти глаголы — [«select»](https://ru.wikipedia.org/wiki/Select_(SQL)) («выбрать»), «[insert»](https://ru.wikipedia.org/wiki/Insert_(SQL)) («вставить»), «[update»](https://ru.wikipedia.org/wiki/Update_(SQL)) («обновить»), и «[delete»](https://ru.wikipedia.org/wiki/Delete_(SQL)) («удалить»).

1. **Выполнение стандартных запросов**
   1. **Выборка всех данных из каждой таблицы**

**connect** 'D:\BD\l4\Medshop.fdb' **user** 'SYSDBA' password 'masterkey'**;**

**commit;**

**select** **\*** **from** Medd**;**

**select** **\*** **from** Consigment**;**

**select** **\*** **from** Clients**;**

**select** **\*** **from** Orders**;**

**select** **\*** **from** Ord\_med**;**

**select** **\*** **from** Appointment**;**

**select** **\*** **from** Appoint\_med**;**

**select** **\*** **from** Side\_effects**;**

**select** **\*** **from** Constraindications**;**

**select** **\*** **from** Category**;**

**select** **\*** **from** Pre\_Category**;**

**select** **\*** **from** Cat\_pre**;**

**select** **\*** **from** Cat\_app**;**

**select** **\*** **from** Analog**;**

* 1. **Выборка данных из одной таблицы при нескольких условиях**

Выведем названия препарата, наличие рецепта (нет) и граммовку (от 20гр до 50гр) :

**select** Name**,**Prescription**,** Weight **from** Medd **where** Prescription **in** **(**0**)**

**and** Weight **between** 20 **and** 50**;**

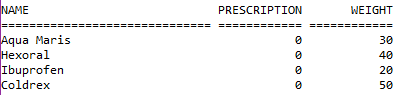


Рис.1. Результат выполнения скрипта

Выведем Поставщика и Сайт, в названии поставщика есть буква ‘с’ или в названии сайта есть ‘ni’:

**select** Provider**,** Site **from** Consigment **where** Provider **like** **(**'%c%'**)**

**or** Site **like** **(**'%ni%'**);**

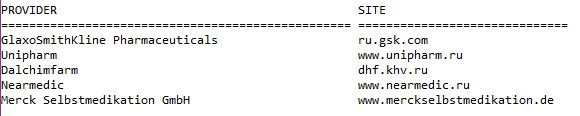


Рис.2. Результат выполнения скрипта

Выведем Покупателя и Скидку, имя покупателя оканчивается на ‘а’ или скидка от 7% до 12%:

**select** Name**,** Discount **from** Clients **where** Name **like** **(**'%a'**)**

**or** Discount **between** 7 **and** 12**;**

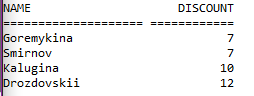


Рис.3. Результат выполнения скрипта

* 1. **Вычисляемое поле в запросе**

Вычислим сумму всех заказов:

**select** **sum(**Summ**)** **from** Orders**;**



Рис.4. Результат выполнения скрипта

* 1. **Выборка всех данных с сортировкой по нескольким полям**

Отсортируем названия и вес препаратов сначала по весу, потом по названию:

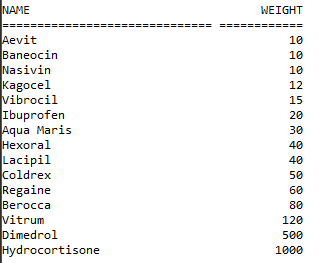


Рис.5. Результат выполнения скрипта

* 1. **Запрос, вычисляющий несколько совокупных характеристик таблиц**

Найдем минимальную, максимальную и среднюю цены среди всех заказов:

**select** **min(**Summ**)** **as** Minimum**,** **avg(**Summ**)** **as** Average**,** **max(**Summ**)** **as** Maximum

**from** Orders**;**



Рис.6. Результат выполнения скрипта

* 1. **Выборка данных из связанных таблиц**

Выведем имя клиента, ID его заказа и сумма заказа, которая больше 120р:

**select** Clients**.**Name**,** Orders**.**ID\_order**,** Orders**.**Summ

**from** Clients**,** Orders

**where** Orders**.**Summ **>**120 **and**

Clients**.**ID\_client**=**Orders**.**ID\_client**;**

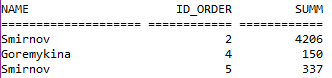


Рис.7. Результат выполнения скрипта

Выведем название категорий и подкатегорий, название которых начинается на букву ‘а’:

**select** Category**.**Name**,** Pre\_Category**.**Name

**from** Cat\_pre**,** Category**,** Pre\_Category

**where** Pre\_Category**.**Name **like** **(**'A%'**)**

**and** Category**.**ID\_cat**=**Cat\_pre**.**ID\_cat

**and** Pre\_Category**.**ID\_precat**=**Cat\_pre**.**ID\_precat**;**

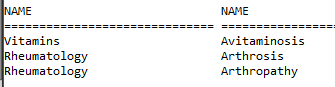


Рис.8. Результат выполнения скрипта

* 1. **Запрос, рассчитывающий совокупную характеристику с использованием группировки**

Сгруппируем название препарата и граммовку, которая больше 20гр:

**select** Name**,** **min(**Weight**)**

**from** Medd **group** **by** Name **having** **min(**Weight**)>**20**;**

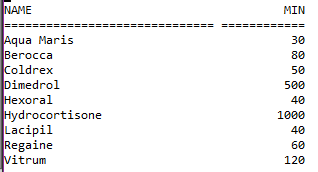


Рис.9. Результат выполнения скрипта

* 1. **Использование вложенного запроса**

Выберем все сроки годности тех препаратов, название поставщиков которое начинается с буквы ‘N’:

**select** Shelf\_life **from** Delivery

**where** ID\_prov **in**

**(select** ID\_cons

**from** Consigment

**where** Provider **like(**'N%'**));**



Рис.10. Результат выполнения скрипта

* 1. **Добавление записей в таблицы**

С помощью оператора INSERT добавим в таблицы по одной записи:

**insert** **into** Side\_effects **values** **(**11**,**'Apathy'**);**

**insert** **into** Medd **values** **(**16**,**'Panadol'**,**0**,**50**);**

* 1. **Измените значений нескольких полей у всех записей, отвечающих заданному условию**

Изменим графу наличие рецепта на 1 (обязательно наличие рецепта) у тех препаратов, вес которых больше 60гр:

**select** **\*** **from** Medd **where** Weight**>**60**;**

**update** Medd **set** Prescription**=**1 **where** Weight**>**60**;**

**select** **\*** **from** Medd **where** Weight**>**60**;**

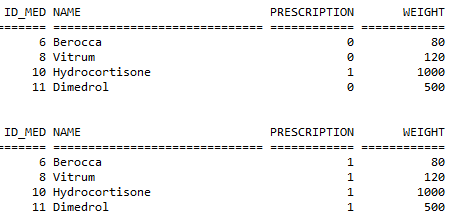


Рис.11. Результат выполнения скрипта

* 1. **Удаление записи, имеющей максимальное (минимальное) значение некоторой совокупной характеристики**

Удалим заказ, у которого раньше всех заканчивается срок годности:

**select** **\*** **from** Delivery**;**

**delete** **from** Delivery **where** Shelf\_life **=** **(select** **min(**Shelf\_life**)** **from** Delivery**);**

**select** **\*** **from** Delivery**;**

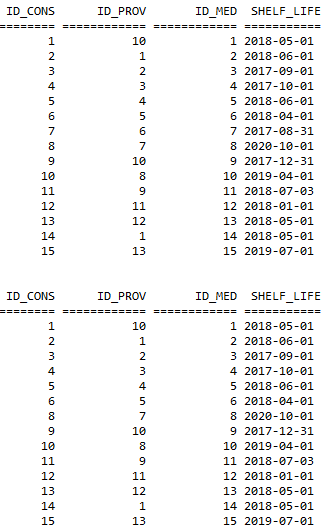


Рис.12. Результат выполнения скрипта

* 1. **Удаление записи в главной таблице, на которые не ссылается подчиненная таблица**

Удалим из списка имя покупателя без заказа:

**select** **\*** **from** Clients**;**

**delete** **from** Clients **where** ID\_client **not** **in**

**(select** ID\_client **from** Orders**);**

**select** **\*** **from** Clients**;**

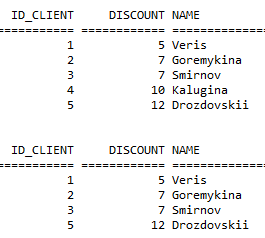


Рис.13. Результат выполнения скрипта

## 5. Индивидуальное задание

1. **Вывести 10 клиентов, которые приобрели наибольшее количество товаров за заданный промежуток времени.**

**select** **first** 10 Clients**.**Name**,** **sum(**Ord\_med**.**Quantity**)** **as** Quantity\_sum

**from** Clients

**natural** **join** Orders

**join** Ord\_Med **on** **(**Orders**.**ID\_order**=**ID\_ord**)**

**where** Date\_ofo **between** '01.12.2016' **and** '03.12.2016'

**group** **by** Clients**.**Name

**order** **by** Quantity\_sum **desc;**

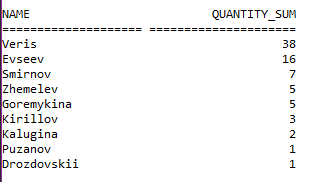


Рис.14. Результат выполнения скрипта

1. **Вывести 3 противопоказания, которые чаще всего указаны в аннотации к лекарствам.**

**select** **first** 3 Name**,** **count(**Const\_med**.**ID\_const**)**

**from** Constraindications

**join** Const\_med **using** **(**ID\_const**)**

**where** Constraindications**.**ID\_const **=** Const\_med**.**ID\_const

**group** **by** Name

**order** **by** **count(**Const\_med**.**ID\_const**)** **desc;**

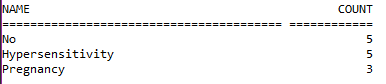


Рис.15. Результат выполнения скрипта

1. **Удалить заказы без содержимого**

**delete** **from** Ord\_med **where** Quantity**=**0**;**

**delete** **from** Orders **where** ID\_order **not** **in**

**(select** ID\_ord **from** Ord\_med**);**

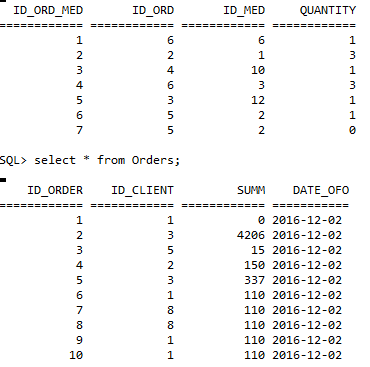


Рис.16. Содержимое таблиц Orders и Ord\_Med до выполнения скрипта

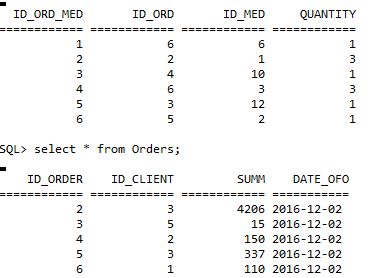


Рис.17. Результат выполнения скрипта

## Сохранение выполненных запросов в виде хранимых процедур

1. **Insert**

**set** term **^** **;**

**create** **procedure** insert\_tables

**as** **begin**

**insert** **into** Medd **values** **(**16**,**'Nasivin'**,**0**,**10**);**

**insert** **into** Consigment **values** **(**14**,**'GlaxoSmithKline Pharmaceuticals'**,**'ru.gsk.com'**);**

**insert** **into** Delivery **values** **(**16**,**10**,**1**,**'01.05.2018'**);**

**insert** **into** Clients **values** **(**13**,**5**,**'Veris'**);**

**insert** **into** Orders **values** **(**14**,**1**,**0**,**'02.12.2016'**);**

**insert** **into** Ord\_med **values** **(**16**,**13**,**10**,**4**);**

**insert** **into** Appointment **values** **(**19**,**'Seasikness'**);**

**insert** **into** Appoint\_med **values** **(**24**,**9**,**15**);**

**insert** **into** Constraindications **values** **(**12**,**'No'**);**

**insert** **into** Const\_med **values** **(**27**,**11**,**15**);**

**insert** **into** Side\_effects **values** **(**11**,**'No'**);**

**insert** **into** Eff\_med **values** **(**31**,**3**,**15**,**25**);**

**insert** **into** Category **values** **(**9**,**'SARS'**);**

**insert** **into** Pre\_Category **values** **(**19**,**'Meningitis'**);**

**insert** **into** Cat\_pre **values** **(**19**,**8**,**16**);**

**insert** **into** Cat\_app **values** **(**8**,**3**,**7**);**

**insert** **into** Analog **values** **(**16**,**15**,**2**);**

**end** **^**

1. **Update**

**create** **procedure** updates\_tables

**as** **begin**

**update** Medd **set** Prescription**=**1**,** Name**=**Name **||** 's' **where** Weight**>**60**;**

**end** **^**

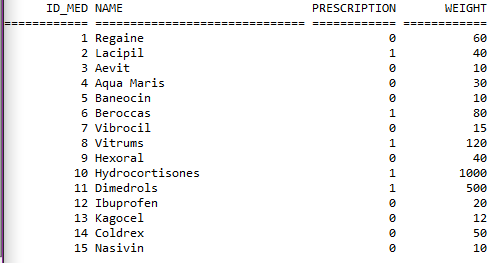


Рис.18. Результат выполнения скрипта

1. **Delete**

**create** **procedure** del\_tables

**as** **begin**

**delete** **from** Delivery **where** Shelf\_life **=** **(select** **min(**Shelf\_life**)** **from** Delivery**);**

**delete** **from** Clients **where** ID\_client **not** **in** **(select** ID\_client **from** Orders**);**

**end** **^**

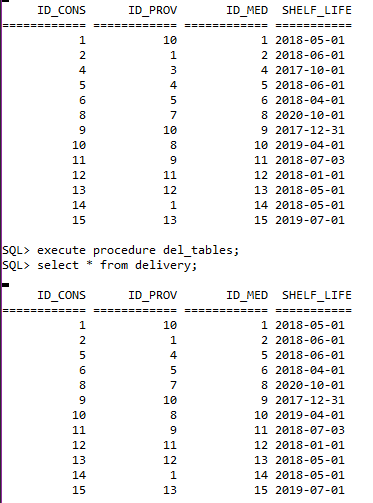


Рис.19-20. Результат выполнения скрипта

1. **Вывод**

Язык DML позволяет достаточно просто выполнять простейшие запросы на модификацию записей, выборку данных, группировку и вычисление совокупных характеристик, операции объединения и сортировку.

Поскольку представленная база данных имеет сложную нормализованную структуру, проводить операции добавления, удаления и редактирования данных вручную слишком трудоемко. Поэтому были реализованы хранимые процедуры, упрощающие эти действия.