

### Практическое задание 3 «SQL сервер, 1 часть»

Описание задания здесь: <http://cmcmsu.info/download/model.sql.interpreter.2005.pdf>

В рамках практического задания 3 необходимо реализовать следующие части сервера и клиента:

- хранение базы данных на диске, чтение, запись, обновление
- взаимодействие сервера с клиентами через сокеты
- пользовательский интерфейс

### Практическое задание 4 «SQL сервер, 2 часть»

Описание задания здесь: <http://cmcmsu.info/download/model.sql.interpreter.2005.pdf>

В рамках практического задания 4 необходимо полностью реализовать SQL сервер, включая интерпретатор подмножества языка SQL, описанного в задании.

Синтаксический анализатор языка должен быть реализован только при помощи метода рекурсивного спуска. В readme должна быть написана грамматика подмножества языка SQL, к которой применим метод рекурсивного спуска.

Синтаксический анализатор должен быть построен строго по этой грамматике.

## **0. Про посылку**

Чтобы получить исполняемые файлы:

*make*

Сначала запускается сервер (либо фоном, либо в другом окне терминала)

*./server*

Затем запускается клиенты(если обрубить всех клиентов(^C) сервер выключается)

*./client*

Для теста из методички

*./client < test.txt*

Все инструкции выводятся клиенту на экран. Чтобы завершить работу, клиент должен ввести слово stop(в этом случае сервер надо обрубать ^C)

Приммер грамматики можно посмотреть в тестах  
**(все 4 команды вводятся в терминал вручную)**

## **I .Реализация**

### **Файл socket\_wrap.hpp**

#### **1.1 class BaseUnSocket**

Базовый класс для сокетов в рамках одной unix машины.

#### **1.2 class UnClientSocket**

Класс, наследуемый от класса BaseUnSocket, предназначен для клиента

#### **1.3 class UnServerSocket**

Класс - наследник класса BaseUnSocket, предназначенный для сервера

#### **1.4 class Exception (в namespace ModelSQL)**

Класс предназначен для удобного представления потенциальных ошибок с сокетам

### **Файл TableWrap.hpp**

#### **2.1 class Table w**

Класс – оболочка для таблицы, реализованной в библиотеке table.c  
Все описания функций в комментариях

## **Файл table.c**

Библиотека для работы с таблицей, взятая с сайта ая  
Таблица немного переделана мной, так как я считаю ее так себе)  
Но, к сожалению, времени на свою не хватает.

## **Файл Scanner.hpp**

### **4.1 class Lex**

Класс – лексема, которую выдает лексический анализатор

### **4.2 class Ident**

Класс - идентификатора (по сути просто является именем  
идентификатора )

### **4.3 class Table ident**

Таблица идентификаторов

### **4.4 class Scanner**

Лексический анализатор, все пояснения есть в комментариях  
файла

## **Файл Parser.hpp**

### **5.1 class Parser**

Читать комментарии в коде

## **Файл Poliz.cpp**

Хранит реализацию функций, манипулирующих с полизом, который  
хранится в парсере