

# База данных "Нагрузка"

## 1 Структура проекта

Проект состоит из трех каталогов: `server`, `client`, `test`. В каталоге `server` содержится исходный код для сервера, в `client` скрипт для запуска клиента, а в `test` находятся наборы тестовых данных и скрипт для тестирования.

## 2 Сборка и запуск

Для сборки сервера необходимо зайти в каталог `server` и выполнить команду `make`. Для удаления созданных объектных файлов используется команда `make clean`.

Запуск сервера осуществляется с помощью команды:

```
python3 server.py [host ip address] [port]
```

(запуск производится в каталоге `server`) Оба аргумента командной строки обязательны. Для завершения работы сервера необходимо использовать `Ctrl+C`.

Запуск клиента производится через команду:

```
python3 client.py [server ip address] [port]
```

(запуск производится в каталоге `client`).

## 3 Запуск тестов

Для запуска тестов необходимо зайти в каталог `test`. И в командной строке ввести:

```
python3 tester.py 5 testfile
```

Изначально в папке содержится 6 файлов с запросами, которые отправляют клиенты (файлы `0.test` - `5.test`) и файл **testfile** в котором содержится сгенерированная автоматически база данных, которую загружает первый клиент(ему соответствует файл с именем `0.txt`). Первый аргумент скрипта `tester.py` - количество клиентов, которые делают запросы к базе данных(нулевой клиент, загружающий базу данных из файла, не

считается), второй параметр - файл с загружаемой базой данных.

В процессе тестирования запускается 5 клиентов и один сервер. Когда все клиенты отработают, сервер завершает работу, также создаются файлы, в которые клиенты записывают результат запроса к базе (файлы с именами 0.test\_res - 5.test\_res)

## 4 Описание команд используемых БД

Все команды, которые подаются на вход базе данных являются **однострочными**.

**addCourse [название курса] [описание нагрузок(возможно с пробельными символами)]** - добавление нового курса. **Каждому курсу присваивается уникальный номер, нумерация с 0.** Описание нагрузок в строгом формате - после каждого наименования учебной нагрузки следует целое число - кол-во часов для данного вида нагрузки. Имена нагрузок изначально не определены, поэтому пользователь может добавлять различные виды нагрузок в базу данных. (Пример: addCourse course1 lectures 10 hw 15).

**deleteCourse index/name [index/name]** - удаление курса по индексу или по названию курса. Если после наименования команды идет ключевое слово index, то после слова index должно идти целое число - номер нужного курса, а после ключевого слова name - строка - имя нужного курса. Нумерация курсов начинается с 0.  
(Пример 1: deleteCourse index 0; Пример 2: deleteCourse name course1)

**updateCourse index/name [index/name] [описание нагрузок(возможно с пробельными символами)]** - изменение нагрузки на определенном курсе (описание курса аналогично описанию курса при использовании deleteCourse).

**updateCourseName [index] [новое имя курса]** - изменение имени курса по индексу курса.

**addWorker [название факультета] [название кафедры] [должность] [имя работника]** - добавление нового работника. Все аргументы обязательны и должны идти в строгом порядке (Названия должностей, факультетов и должностей не определены изначально, что дает возможность пользователю брать любые названия должностей, кафедр и факультетов. **Каждое название - одно слово без пробельных символов**).

**uploadFromTxtFile [имя текстового файла]** - загрузка базы данных из файла (Хранение данных производится в виде команд для БД).

**saveAsTxtFile** [имя текстового файла] - сохранение базы данных в файл.

**deleteWorker** [факультет] [кафедра] [должность] [имя] - удаление сотрудника из базы данных. Для большей гибкости добавлена возможность удаления нескольких сотрудников с помощью символа !. Если один из 4х обязательных параметров неопределен, то можно указать вместо него ! и тогда операция будет применена ко всем работникам, которые соответствуют указанному шаблону. (Пример: в базе было 2 сотрудника: **fac1 kaf1 post1 name1** и **fac1 kaf2 post2 name2**. Если указать команду **deleteWorker fac1 kaf2 post2 name2** или команду **deleteWorker fac1 kaf2 !** !, то будет удален второй работник, а первый останется в базе, а если применить команду **deleteWorker ! ! !** !, то будут удалены оба сотрудника).

**deleteWorkersCourse** [факультет] [кафедра] [должность] [имя] [номер курса] - удаление курса у сотрудника соответствующего шаблону (аналогичного шаблону в команде **deleteWorker**).

**updateWorkersCourse** [факультет] [кафедра] [должность] [имя] [номер курса] [строка (возможно с пробельными символами)] - изменение курса у сотрудника соответствующего шаблону (аналогичного шаблону в команде **deleteWorker**).

**addWorkersCourse** [факультет] [кафедра] [должность] [имя] [номер курса] - добавление нового курса сотруднику, который соответствует шаблону (аналогичного шаблону в команде **deleteWorker**).

**getWorkersCourse** [факультет] [кафедра] [должность] [имя] - вывод на консоль всех курсов сотрудника, который соответствует шаблону (аналогичного шаблону в команде **deleteWorker**).

**select columns** [названия в иерархии сотрудников(только **faculty**, **department**, **post**) или наименования нагрузок(например: **lectures**, **hw**)] **end \_columns rows \_for \_all** [наименования в иерархии(например: **professor**)] **sum \_row/end \_row** - выборка из базы данных по параметрам, которая отображается в виде таблицы в текстовом формате. После ключевого слова **columns** необходимо указать те критерии, которые будут отображаться по столбцам. Поле имени указывать не нужно, оно добавляется автоматически. Отбор осуществляется по принципу объединения имен иерархии указанных в поле после **rows \_for \_all**. **sum \_row** означает, что необходимо в конце таблицы вывести дополнительно строку, в которой указаны просуммированные данные по столбцам (если столбец содержит информацию в виде чисел), **end \_row** значит, что суммирование по столбцам производиться не будет.

Пример: **select columns faculty lectures end \_columns rows \_for \_all docent professor sum\_row** - Выведет таблицу из 3х столбцов (в первом имя сотрудника, во втором название факультета, в третьем сумму часов лекций, которые проводит этот сотрудник.) В каждой строке будет указан один сотрудник, а в последней строке будет указано суммарное кол-во часов лекций у всех сотрудников из выборки.

**exit** - команда для завершения работы клиента (он отключается от сервера).