**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

Индивидуальное Домашнее Задание

**по дисциплине «Введение в нереляционные базы данных»**

**Тема: Электронный журнал успеваемости**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 6383 |  | Азаревич А.Д. |
|  |  | Гомонова А.А. |
| Студентка гр. 6382 |  | Зацепилова С.В. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2019

|  |  |  |
| --- | --- | --- |
| Студенты  Азаревич А.Д.  Гомонова А.А.  Зацепилова С.В. | | |
| Группы 6383, 6382 | | |
| Тема проекта: Электронный журнал успеваемости Neo4j | | |
| Исходные данные:  Необходимо реализовать приложение электронный журнал успеваемости для СУБДNeo4j. | | |
| Содержание пояснительной записки:  «Содержание»  «Введение»  «Сценарии использования»  «Модель данных»  «Разработанное приложение»  «Заключение»  «Список использованных источников»  «Приложение» | | |
| Предполагаемый объем пояснительной записки:  Не менее 15 страниц. | | |
| Дата выдачи задания: | | |
| Дата сдачи реферата: | | |
| Дата защиты реферата: | | |
| Студенты гр.6383 |  | Азаревич А.Д. |
|  |  | Гомонова А.А. |
| Студентка гр.6382 |  | Зацепилова С.В. |
| Преподаватель |  | Заславский М.М. |

**ЗАДАНИЕ**

**Аннотация**

В рамках данного курса предполагалось разработать какое-либо приложение в команде на одну из поставленных тем. Была выбрана тема электронный журнал ведомости для СУБД Neo4j.

Приложение электронный журнал позволит добавлять и удалять студентов, преподавателей, ставить оценку студенту, добавлять и удалять предметы и направление обучения, назначать предметы преподавателем. С помощью приложения можно получить следующие статистики: «Средняя оценка на факультете», «Оценки, выставляемые преподавателем», «Загруженность преподавателей на кафедре».

**Summary**

As part of this course, it was supposed to develop any application in the team on one of the topics posed. The topic was selected as the electronic journal sheet for the Neo4j DBMS.

The electronic journal application will allow you to add and remove students, teachers, rate the student, add and remove subjects and the direction of study, assign subjects as a teacher. Using the application, you can get the following statistics: "Average grade at the faculty", "Grades given by the teacher", "Workload of teachers in the department."

**Содержание**

[ВВЕДЕНИЕ 5](#_Toc27497933)

[1. СЦЕНАРИЙ ИСПОЛЬЗОВАНИЯ 6](#_Toc27497934)

[1.1. Макеты пользовательского интерфейса 6](#_Toc27497935)

[1.2. Описание сценариев использования 8](#_Toc27497936)

[1.2.1. Просмотр главной страницы, импорт и экспорт БД 8](#_Toc27497937)

[1.2.2. Переход на страницу с информацией о преподавателях и предметах, выполнение запросов на странице 9](#_Toc27497938)

[1.2.3. Переход на страницу с информацией о факультетах, выполнение запросов на странице 10](#_Toc27497939)

[1.2.4. Переход на страницу с информацией о кафедрах, выполнение запросов на странице 10](#_Toc27497940)

[1.2.5. Переход на страницу с информацией о студентах, выполнение запросов на странице 11](#_Toc27497941)

[1.2.6. Переход на страницу статистических запросов, выполнение запросов 12](#_Toc27497942)

[2. МОДЕЛЬ ДАННЫХ 13](#_Toc27497943)

[2.1. NoSQL модель данных (Neo4j) 13](#_Toc27497944)

[2.1.1. Графическое представление 13](#_Toc27497945)

[2.1.2. Описание назначений коллекций, типов данных и сущностей 13](#_Toc27497946)

[2.1.3. Расчёт объема 15](#_Toc27497947)

[2.1.4. Примеры запросов 16](#_Toc27497948)

[2.2. SQL модель данных 16](#_Toc27497949)

[2.2.1. Графическое представление 16](#_Toc27497950)

[2.2.2. Описание назначений коллекций, типов данных и сущностей 16](#_Toc27497951)

[2.2.3. Расчёт объёма 18](#_Toc27497952)

[2.2.4. Примеры запросов 19](#_Toc27497953)

[2.3. Сравнение Neo4j и SQL моделей данных 20](#_Toc27497954)

[3. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ 20](#_Toc27497955)

[3.1. Краткое описание 20](#_Toc27497959)

[3.2. Схема экранов приложения 21](#_Toc27497960)

[3.3. Использованные технологии 22](#_Toc27497961)

[ЗАКЛЮЧЕНИЕ 23](#_Toc27497962)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc27497963)

[ПРИЛОЖЕНИЕ А. ДОКУМЕНТАЦИЯ ПО СБОРКЕ И РАЗВЕРТЫВАНИЮ ПРИЛОЖЕНИЯ 25](#_Toc27497964)

# ВВЕДЕНИЕ

Цель работы – создать приложение электронный журнал успеваемости, с помощью которого можно вести учет успеваемости каждого студента, выполнять различные запросы с целью получения необходимой информации.

Было решено разработать веб-приложение, которое позволит хранить информацию о факультетах, группах, студентах, преподавателях и оценках, полученных студентами.

Качественные требования к решению: требуется разработать приложение с использованием СУБД Neo4j.

# СЦЕНАРИЙ ИСПОЛЬЗОВАНИЯ

## Макеты пользовательского интерфейса

Главная страница представлена на рис. 1, страница с информацией о студентах на рис. 2, рис. 3, страница с информацией о преподавателях на рис. 4, страница статистических запросов представлена на рис. 5, страница факультетов на рис. 6, страница кафедр представлена на рис. 7.

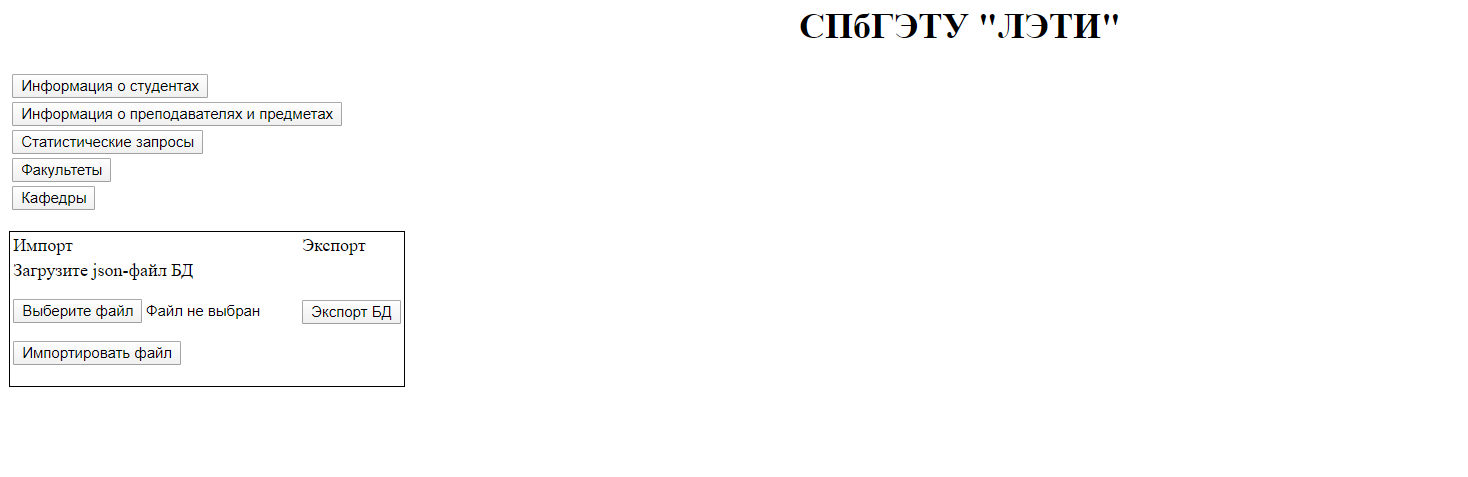


Рисунок 1 – Главная страница приложения

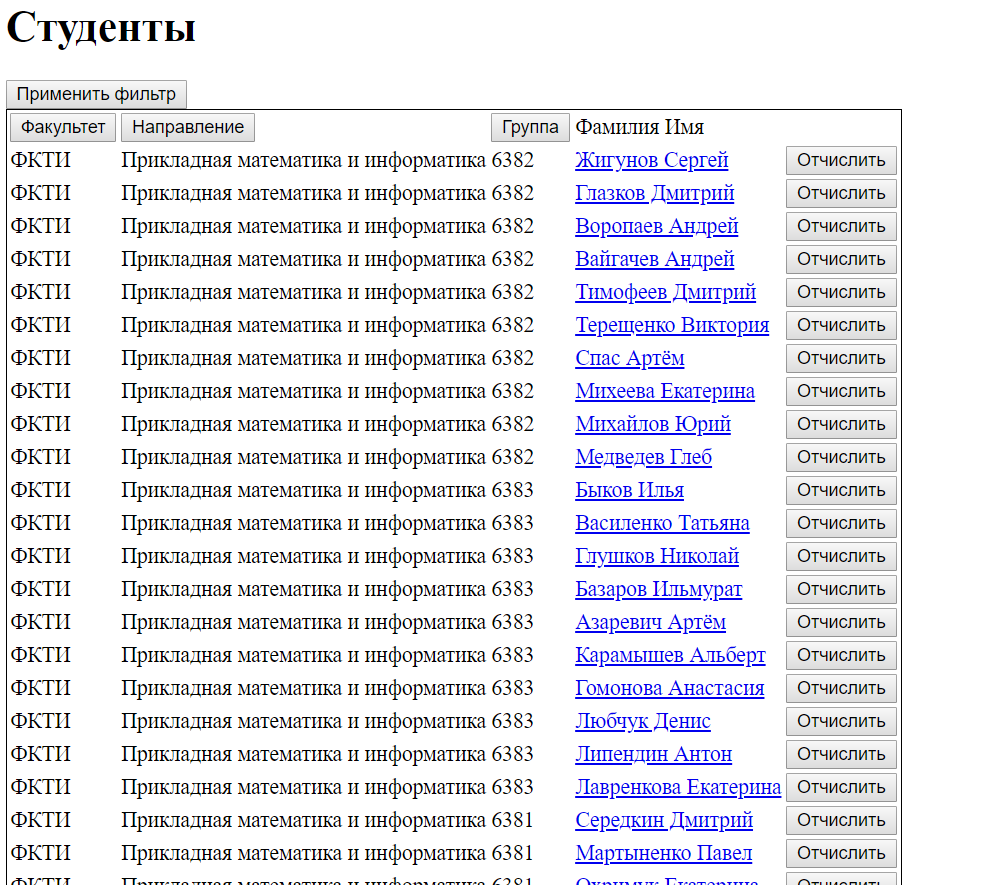


Рисунок 2- Страница с информацией о студентах 1

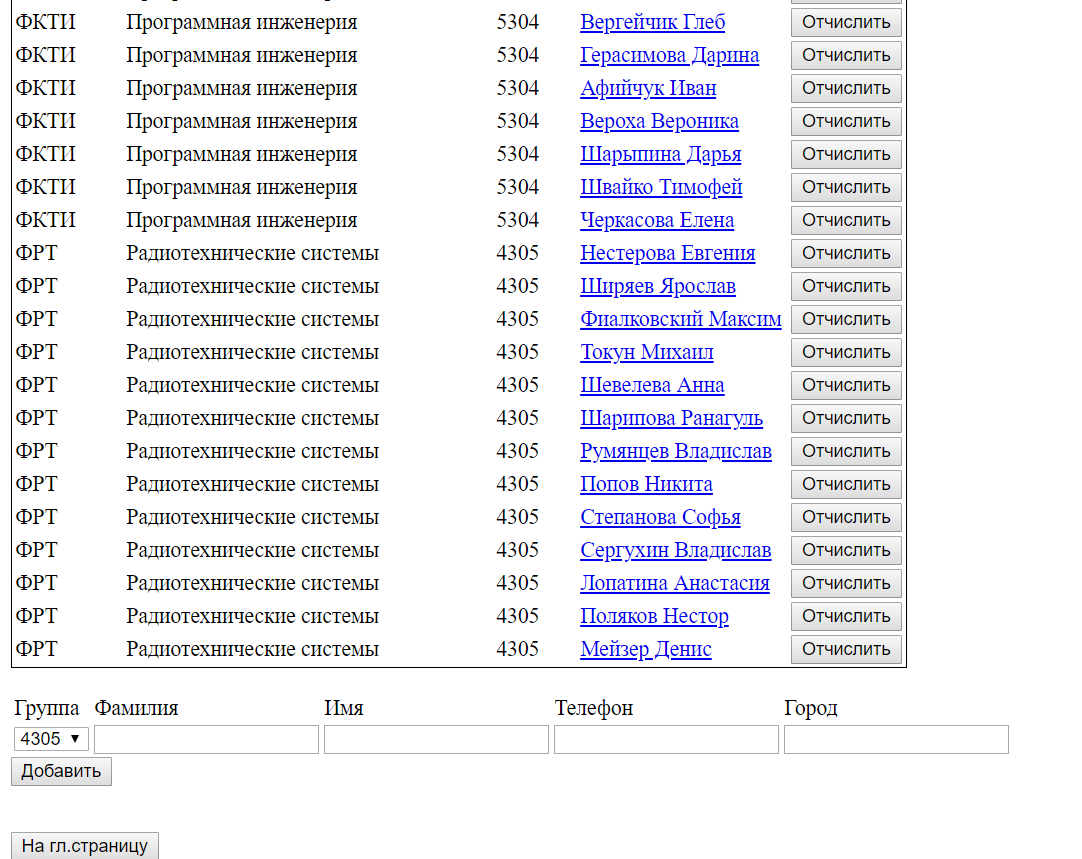


Рисунок 3 - Страница с информацией о студентах2

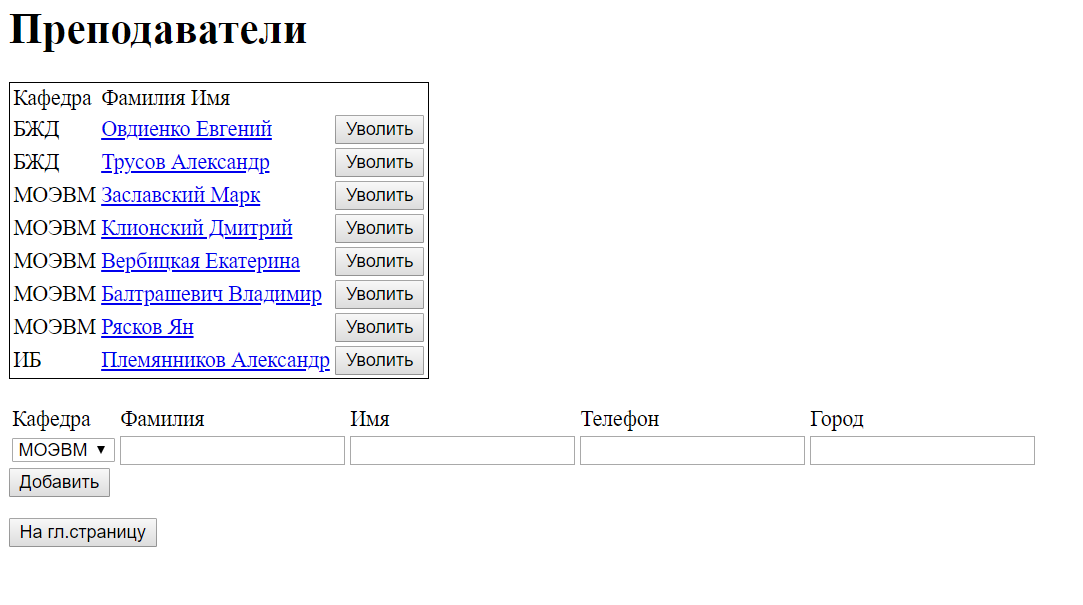


Рисунок 4 – Страницас информацией о преподавателях

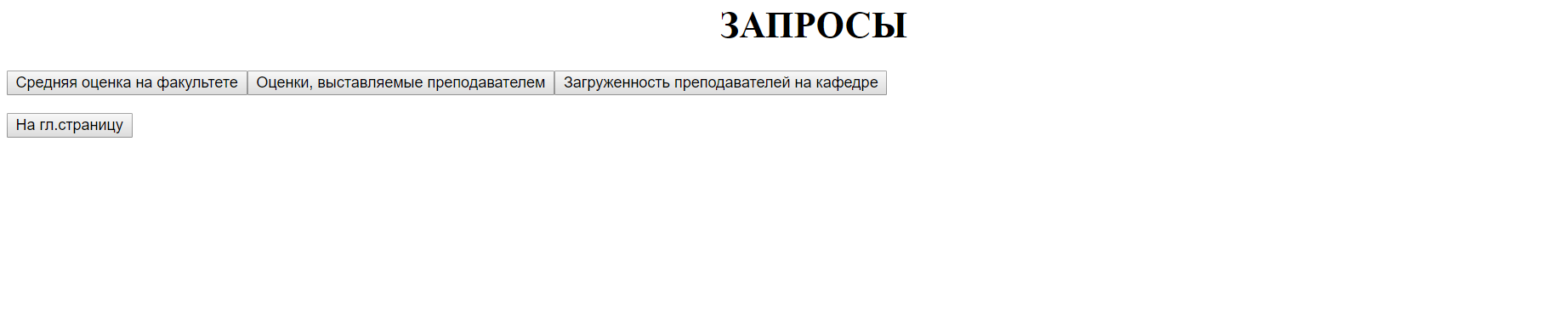


Рисунок 5–Страница статистических запросов

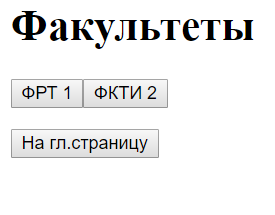


Рисунок 6 - Страница факультетов

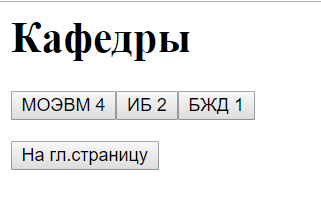


Рисунок 7 - Страница кафедр

## Описание сценариев использования

### Просмотр главной страницы, импорт и экспорт БД

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование** | **Описание сценария** | **Ожидаемый результат** |
| 1 | Просмотр главной страницы | Открыть главную страницу | Отображаются 8 кнопок:   1. Кнопка для перехода на страницу информации о студентах; 2. Кнопка для перехода на страницу с информацией о преподавателях и предметах; 3. Кнопка для перехода на станицу статистических запросов; 4. Кнопка для перехода на страницу факультетов; 5. Кнопка для перехода на страницу кафедр; 6. Кнопка выбора файла с пользовательского компьютера; 7. Кнопка «Импортировать файл» для для осуществления операции импорта БД; 8. Кнопка «Экспорт БД» для осуществления операции экспорта БД. |
| 2 | Импорт данных | В окошке импорта/экспорта нажать на кнопку «Выберите файл» | Открывается окно выбора файла с пользовательского компьютера. |
| Выбрать файл на пользовательском компьютере, нажать на кнопку «Открыть». | Окно выбора файла с пользовательского файла закрылось, выбранный файл отображается в окошке импорта. |
| Нажать на кнопку «Импортировать файл» | Данные импортированы. |
| 3 | Экспорт данных | В окошке импорта/экспорта нажать на кнопку «Экспорт БД» | Выполнено скачивание файла данных в формате .json. |

### Переход на страницу с информацией о преподавателях, выполнение запросов на странице

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование** | **Описание сценария** | **Ожидаемый результат** |
| 1 | Переход на страницу с информацией о преподавателях | На главной странице нажать на кнопку «Информация о преподавателях и предметах» | Открыта страница с информацией о преподавателях и предметах, доступны кнопки «Уволить», «Добавить», «На главную страницу». |
| 2 | Увольнение преподавателя | Выбрать любого преподавателя и нажать на кнопку «Уволить» | Выбранный преподаватель не отображается в списке преподавателей. |
| 3 | Добавление преподавателя | Нажать на выпадающий список «Кафедры», выбрать необходимую кафедру | В поле «Кафедра» отображается выбранная кафедра. |
| Заполнить поля «Фамилия», «Имя», «Телефон», «Город», нажать на кнопку «Добавить» | В списке преподавателей отображается добавленный преподаватель, справа от преподавателя доступна кнопка «Уволить». |
| 4 | Просмотр информации о нагрузке преподавателя | Нажать на Фамилию Имя преподавателя | Открыта страница с информацией о нагрузке преподавателя, доступны кнопки «Добавить нагрузку», «Освободить нагрузку», «На гл. страницу», «Назад». |
| 5 | Удаление/добавление нагрузки преподавателя | Нажать на кнопку «Освободить» в строке предмета, который необходимо снять с преподавателя | Предмет в таблице не отображается. |
| В выпадающем списке выбрать предмет, который необходимо добавить, нажать на кнопку «Добавить нагрузку» | Выбранный предмет отображается в таблице предметов, которым обучает преподаватель. |
| 6 | Переход на главную страницу | Нажать на кнопку «На гл. страницу» | Выполнен переход на главную страницу. |

### Переход на страницу с информацией о факультетах, выполнение запросов на странице

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование** | **Описание сценария** | **Ожидаемый результат** |
| 1 | Переход на страницу с информацией о факультетах | На главной странице нажать на кнопку «Факультеты» | Открыта страница «Факультеты», на странице отображаются существующие факультеты, доступна кнопка «На гл. страницу». |
| 2 | Переход на страницу с информацией о выбранном факультете | Нажать на наименование факультета | Открыта страница с информацией о факультете, отображается список направлений на выбранном факультете и количество групп. Доступны кнопки «Добавить направление», «На гл. страницу», «Назад». |
| 3 | Добавление направления | В поле «Добавить направление» ввести наименование направления, нажать на кнопку «Добавить направление» | Добавленное направление отображается в списке направлений, справа от направления доступна кнопка «Убрать». |
| 4 | Просмотр информации о направлении | Нажать на наименование направления | Открыта страница с информацией о выбранном направлении, доступны кнопки «Добавить группу», «На гл. страницу», «Назад». |
| 5 | Добавление группы | В поле «Добавить группу» ввести номер группы, которую необходимо добавить, нажать на кнопку «Добавить группу» | Добавленная группа отображается в списке групп, справа от группы доступна кнопка «Убрать». |
| 6 | Удаление группы | Справа от группы нажать на кнопку «Убрать» | Добавленная группа в списке групп не отображается. |
| 7 | Удаление направления | Справа от добавленного направления нажать на кнопку «Убрать» | Добавленное направление в списке направлений не отображается. |
| 8 | Переход на страницу с информацией о факультетах | Нажать на кнопку «Назад» | Открыта страница «Факультеты», на странице отображаются существующие факультеты, доступна кнопка «На гл. страницу». |

### Переход на страницу с информацией о кафедрах, выполнение запросов на странице

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование** | **Описание сценария** | **Ожидаемый результат** |
| 1 | Переход на страницу с информацией о кафедрах | На главной странице нажать на кнопку «Кафедры» | Открыта страница «Кафедры», на странице отображаются существующие кафедры, кнопка «На гл. страницу». |
| 2 | Переход на страницу с информацией о выбранной кафедре | Нажать на наименование кафедры | Открыта страница с информацией о выбранной кафедре, отображается список направлений, соответствующие направлению предметы и количество связей у каждого предмета, доступна кнопка «Добавить предмет». |
| 3 | Добавление предмета | В поле «Добавить предмет» ввести наименование предмета и нажать на кнопку «Добавить предмет» | Предмет отображается в списке предметов, справа от наименования предмета доступна кнопка «Убрать». |
| 4 | Удаление предмета | Нажать на кнопку «Убрать» | Выбранный предмет в списке предметов не отображается. |
| 5 | Переход на страницу с информацией о кафедрах | Нажать на кнопку «Назад» | Открыта страница «Кафедры», на странице отображаются существующие кафедры, кнопка «На гл. страницу». |

### Переход на страницу с информацией о студентах, выполнение запросов на странице

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование** | **Описание сценария** | **Ожидаемый результат** |
| 1 | Переход на страницу с информацией о студентах | На главной странице нажать на кнопку «Информация о студентах» | Открыта страница с информацией о студентах, отображается список студентов со следующей информацией: «Факультет», «Направление», «Группа», «Фамилия Имя», доступна кнопка «Применить фильтр», «Добавить», «На гл. страницу», «Отчислить». |
| 2 | Добавление студента | В выпадающем списке выбрать группу, ввести Фамилию, Имя, телефон, город, нажать на кнопку «Добавить» | Добавленный студент отображается в списке студентов, справа от студента отображается кнопка «Отчислить». |
| 3 | Удаление студента | Справа от студента нажать на кнопку «Отчислить» | Добавленный студент в списке не отображается. |
| 4 | Фильтр по факультетам | Нажать на кнопку «Факультет» | Отображается список факультетов. |
| Выбрать факультет(-ы), по которому(-ым) требуется применить фильтр, нажать на кнопку «Применить фильтр» | В списке отображается студенты, обучающиеся на выбранном факультете(-ах). |
| 5 | Фильтр по направлениям | Нажать на кнопку «Направление» | Отображается список направления. |
| Выбрать направление(-я), по которому(-ым) требуется применить фильтр, нажать на кнопку «Применить фильтр» | В списке отображается студенты, обучающиеся на выбранном направлении(-ях). |
| 6 | Фильтр по группе | Нажать на кнопку «Группа» | Отображается список групп. |
| Выбрать группу(-ы), студенты которой(-ых) должны отобразиться, нажать на кнопку «Применить фильтр» | В списке отображается студенты выбранной(-ых) группы (групп). |
| 7 | Просмотр зачетки студента | Нажать на Фамилию Имя студента | Открыта страница с информацией о студенте, отображается информация об оценках студента по каждому предмету, доступны кнопки «Удалить отметку», «Выставить». |
| 8 | Удаление оценки | Нажать на кнопку «Удалить отметку» напротив выбранного предмета | Строка с оценкой по предмету удалена. |
| 9 | Выставление оценки | Заполнить поля с информацией о предмете, преподавателе, дата сдачи и оценка, нажать на кнопку «Выставить» | Отображается строка в таблице успеваемости с соответствующей информацией о выставленной оценке. |

### Переход на страницу статистических запросов, выполнение запросов

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование** | **Описание сценария** | **Ожидаемый результат** |
| 1 | Переход на страницу статистических запросов | На главной странице нажать на кнопку «Статистические запросы» | Отображаются доступные запросы: «Средняя оценка на факультете», «Оценки, выставляемые преподавателем», «Загруженность преподавателей на кафедре». |
| 2 | Запрос «Средняя оценка на факультете» | На странице статистических запросов нажать на кнопку «Средняя оценка на факультете» | Открыта страница запроса, доступен выбор параметров запроса и кнопка «Выполнить». |
| Заполнить необходимые параметры запроса, нажать на кнопку «Выполнить» | Отображается результат запроса. |
| 3 | Запрос «Оценки, выставляемые преподавателем» | На странице статистических запросов нажать на кнопку «Оценки, выставляемые преподавателем» | Открыта страница запроса, доступен выбор параметров запроса и кнопка «Выполнить». |
| Заполнить необходимые параметры запроса, нажать на кнопку «Выполнить» | Отображается результат запроса. |
| 4 | Запрос «Загруженность преподавателей на кафедре» | На странице статистических запросов нажать на кнопку «Загруженность преподавателей на кафедре» | Открыта страница запроса, доступен выбор параметров запроса и кнопка «Отправить запрос». |
| Заполнить необходимые параметры запроса, нажать на кнопку «Отправить запрос» | Отображается результат запроса. |

# МОДЕЛЬ ДАННЫХ

## NoSQL модель данных (Neo4j)

### Графическое представление

Графическое представление модели данных Neo4j показано на рис. 8.

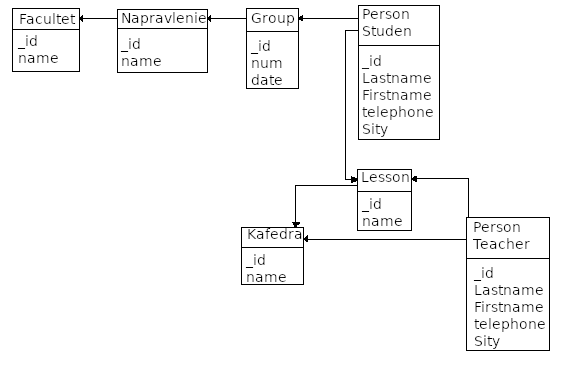


Рисунок 8 – Графическое представление модели данных Neo4j

### Описание назначений коллекций, типов данных и сущностей

Модель состоит из 6 сущностей:

1. Факультет:

* "\_id", int - идентификатор. *4B*
* "name", String - название факультета. *4\*2B*

*Итого: 12В*

1. Направление обучения:

* "\_id", int - идентификатор. *4B*
* "name", String - Название направления. *15\*2B*

*Итого: 34В*

1. Группа:

* "\_id", int - идентификатор. *4B*
* "num", String - Номер группы. *4\*2B*

*Итого: 12В*

1. Человек (может быть как студентом, так и преподавателем):

* "\_id", int - идентификатор. *4B*
* "Lastname", String - Фамилия. *8\*2B*
* "Firstname", String - Имя. *6\*2B*
* "telephone", String - Номертелефона. *14\*2B*
* "Sity", String - Город. *10\*2B*;

*Итого: 80В*

1. Кафедра:

* "\_id", int - идентификатор. *4B*
* "name", String - Название кафедры. *4\*2B*

*Итого: 12В*

1. Предмет:

* "\_id", int - идентификатор. *4B*
* "name", String - Название предмета. *7\*2B*

*Итого: 18В*

Существующие связи между сущностями:

* Направление->Факультет ("Данной специальности учат на данном факультете"). Полей нет.
* Группа->Направление ("Данная группа обучается по этому направлению"). Полей нет.
* Человек(студент)->Группа ("Студент зачислен в данную группу"). Полей нет.
* Человек(Студент)->Предмет ("Студент имеет в зачётке отметку по предмету"). Поля:
  + - * "\_id", int - идентификатор. *4B*
      * "assessment", char - Полученная оценка. *2В*
      * "date", String - Дата выставления оценки. *10\*2В*
      * "teach\_id", int - Идентификатор преподавателя. *4В*

*Итого: 30В*

* Человек(Преподаватель)->Кафедра ("Преподователь работает на кафедре"). Полей нет.
* Человек(Преподаватель)->Предмет ("Преподователь ведё"). Полей нет.
* Предмет->Кафедра ("Предмет изучается на кафедре"). Полей нет.

### Расчёт объема

Предположим, что существует *F* факультетов, на каждом факультете происходит обучение в среднем по 4-м направлениям, на каждом направлении сформировано в среднем по 3 группы, в каждой из которых в среднем обучается по 15 человек, каждый из которых сдал по 3 предмета. Так же предположим, что есть на каждом факультете присутствует в среднем по 3 кафедры, на которых в среднем преподают по 5 преподавателей. На каждой кафедре в среднем проходят 3 предмета.

"Чистый" объём: Объём сущности "Человек" в среднем составляет 80В, сущностей "Факультет", "Группа" и "Кафедра" - 12В, сущности "Направление" - 34В, сущности "Предмет" - 18В. Объём связи "Студент->Предмет" - 30В. Тогда занимаемый объём: \_32290\*F \_байт.

Фактический объём: Объём сущностей:

* "Человек" - 817В
* "Факультет" - 216В
* "Направление" - 503В
* "Группа" - 784В
* "Предмет" - 638В
* "Кафедра" - 216В

Связь "Студент->Предмет" - 713В

Тогда общий объём БД: *558533\*F* байт

Избыточность модели: 558533/32290=17.3

### Примеры запросов

1. Добавить нового преподавателя

MATCH (k:Kafeda{name:"Название"}), (l:Lesson{name:"Название"}) CREATE (p:Person:Teacher{...})-[:work]-(k), (p)-[:teach]->(l);

1. Определить средний бал по предмету "HN" для кадлой группы (которая его проходила)

match (g)--()-[r]-(:Lesson{name:"HN"}) return g.num, avg(toInteger(r.assesment)) as q;

## SQL модель данных

### Графическое представление

Графическое представление модели данных SQL показано на рис. 9.



Рисунок 9 - Графическое представление модели данных SQL

### Описание назначений коллекций, типов данных и сущностей

SQL модель состоит из следующих таблиц:

1. Факультет:

* id, int - уникальный номер записи. *4B*
* "name", String - название факультета. *4\*2B*

*Итого: 12В*

1. Направление обучения:

* "\_id", int - идентификатор. *4B*
* "id\_f", int - "ссылка" на факультет. *4B*
* "name", String - Название направления. *15\*2B*

*Итого: 38В*

1. Группа:

* "\_id", int - идентификатор. *4B*
* "id\_n", int - "ссылка" на факультет. *4B*
* "num", String - Номер группы. *4\*2B*

*Итого: 16В*

1. Человек:

* "\_id", int - идентификатор. *4B*
* "Lastname", String - Фамилия. *8\*2B*
* "Firstname", String - Имя. *6\*2B*
* "telephone", String - Номертелефона. *14\*2B*
* "Sity", String - Город. *10\*2B*;

*Итого: 80В*

* 1. Студент:
* "\_id", int - идентификатор. *4B*
* "id\_p", int - "ссылка" на человека. *4B*

*Итого 8B* 4.2 Преподаватель:

* "\_id", int - идентификатор. *4B*
* "id\_p", int - "ссылка" на человека. *4B*

*ИтогоЖ 8В* 5. Кафедра:

* "\_id", int - идентификатор. *4B*
* "name", String - Название кафедры. *4\*2B*

*Итого: 12В* 6. Предмет:

* "\_id", int - идентификатор. *4B*
* "id\_k", int - на какой кафедре изучается. *4B*
* "name", String - Название предмета. *7\*2B*

*Итого: 22В* 7. Препод\_Предмет:

* "id\_t", int - какой препод. *4B*
* "id\_l", int - какой предмет ведёт. *4B*

*Итого: 8В* 8. Оценки:

* "id\_s", int - Студент. *4B*
* "id\_l", int - по предмету. *4B*
* "id\_p", int - преподаватель. *4B*
* "assessment", char - поставил отметку. *2B*
* "date", String - такого числа. *10\*2B*

\_Итого: 34В"

### Расчёт объёма

Предположим, что существует *F* факультетов, на каждом факультете происходит обучение в среднем по 4-м направлениям, на каждом направлении сформировано в среднем по 3 группы, в каждой из которых в среднем обучается по 15 человек, каждый из которых сдал по 3 предмета. Так же предположим, что есть на каждом факультете в среднем есть по 3 кафедры, на которых в среднем преподают по 5 преподавателей. На каждой кафедре в среднем проходят 3 предмета. В среднем, каждый преподаватель ведёт по 2 предмета.

"Чистый" объём: Средний размер записи для каждой таблицы (в байтах):

* "Факультет" – 12
* "Направление" – 38
* "Группа" – 16
* "Человек" – 80
* "Студент" – 8
* "Преподаватель" – 8
* "Кафедра" – 12
* "Предмет" – 22
* "Отметка" – 34
* "ПреподПредмет" – 4

Тогда общий объём БД: *36350\*F* байт

Фактический объём: Объём записей в таблице (в байтах):

* "Факультет" – 292
* "Направление" – 300
* "Группа" – 297
* "Человек" – 308
* "Студент" – 288
* "Преподаватель" – 288
* "Кафедра" – 288
* "Предмет" – 293
* "Отметка" – 318
* "ПреподПредмет" – 287

Тогда общий объём БД: *299044\*F* байт

Избыточность модели: 299044/36350=8.2

### Примеры запросов

1. Добавить нового преподавателя

INSERT INTO Человек VALUES(...)

INSERT INTO Преподаватель VALUES(...)

INSERT INTO Препод\_Предмет VALUES(...)

1. Определить средний бал по предмету "HN" для кадлой группы (которая его проходила)

SELECT num, avg(Assessment) AS q FROM ( SELECT num, id\_gr, Студенты.id as qq FROM Группы LEFT JOIN Студенты ON Группы.id = Студенты.id\_g ) INNER JOIN Оценки ON qq=Оценки.id\_s GROUP BY id\_gr

## Сравнение Neo4j и SQL моделей данных

Запросы на SQL-языке являются более громоздкими, чем на языке Neo4j Cypher. В реализации реляционной модели данных пришлось бы создать слишком большое количество таблиц для связи всей информации.

Neo4j-БД требует больше памяти по сравнению с SQL-БД для хранения полей. Однако время выполнения запроса в Neo4j-БД в среднем составляет 57 мс, в то время как в SQL-БД – 108 мс.

Исходя из вышеперечисленного, можно сказать, что для рассматриваемой задачи Neo4jподходит в большей мере.

# РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ



## Краткое описание

Для упрощения процесса разработки было принято решение разделить приложение на клиентскую и серверную части.

Клиентская часть реализована с использованием JavaScript, HTML, шаблонизатора Pug и JS библиотек JQuery и AnyСhart.

Серверная часть приложения реализована с использованием Node.js и фреймворка express. Так же использовались следующие фреймворки:

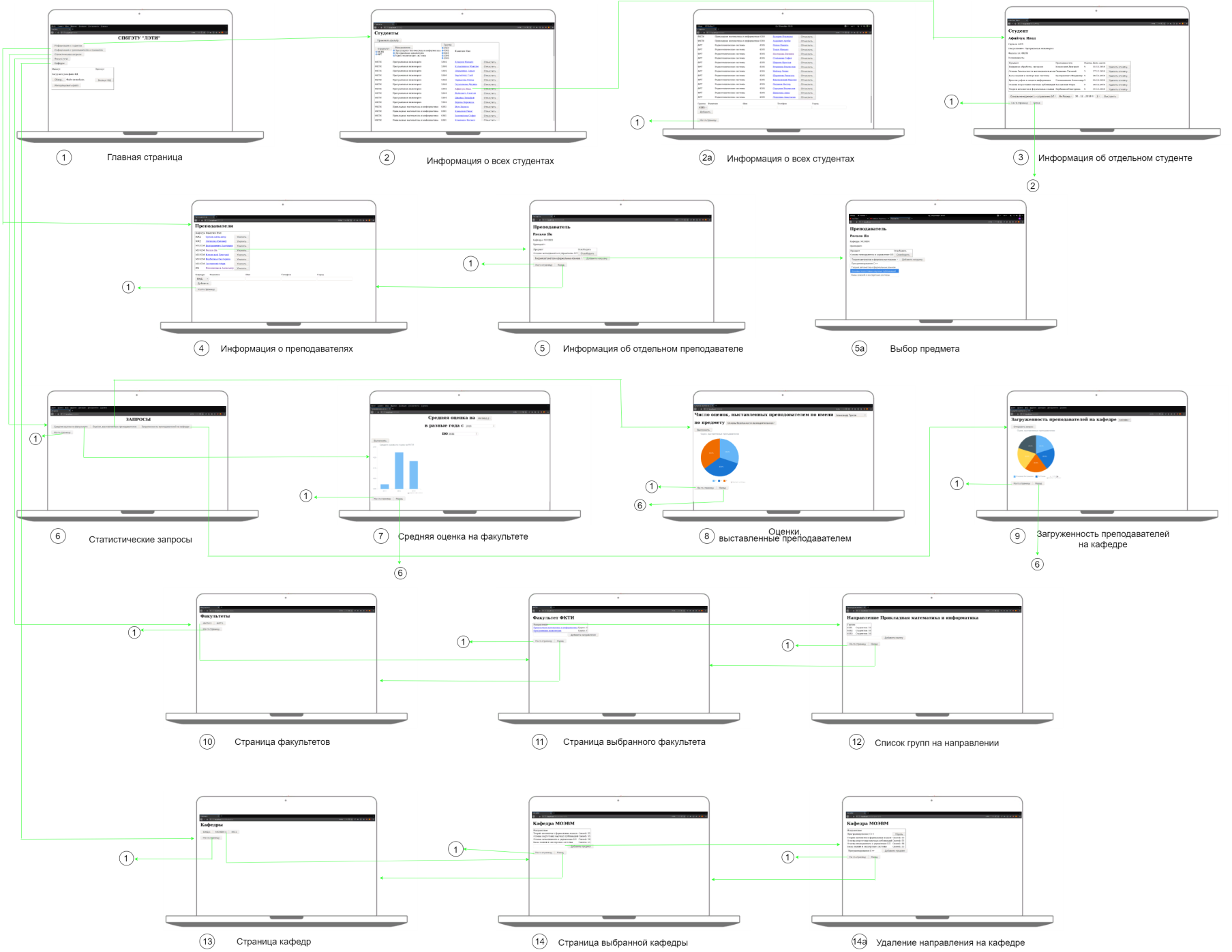
* pug (для генерации HTMLстраниц по заданным шаблонам);
* body-parser (для работы с клиентскими запросами), multer (для возможности загрузки файлов с импортируемыми данными от клиентской части);
* neo4j-driver (для работы с базой данных Neo4j).

Поскольку приложение требует установки стороннего программного обеспечения: Node.js и Neo4j − было решено для упрощения развертывания приложения использовать docker.

Ссылка на приложение доступна в разделе "Список использованных источников": [1]

## Схема экранов приложения

Схема экранов приложения представлена на рис. 10.



## Использованные технологии

БД: Neo4j

Backend: Node.js, express, body-parser, multer, neo4j-driver

Frontend: JS, HTML, Pug, JQuery, AnyСhart

# ЗАКЛЮЧЕНИЕ

В ходе работы было разработано приложение, позволяющее пользователю вести электронный журнал успеваемости, а именно: добавлять и удалять студентов, преподавателей, ставить оценку студенту, добавлять и удалять предметы и направление обучения, назначать предметы преподавателем. С помощью приложения можно получить следующие статистики: «Средняя оценка на факультете», «Оценки, выставляемые преподавателем», «Загруженность преподавателей на кафедре». Также были реализованы функции импорта и экспорта данных.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ссылка на репозиторий: <https://github.com/moevm/nosql2h19-marks-neo4j>
2. Документация Neo4j: <https://neo4j.com/docs/cypher-manual/current/>
3. Документация Apoc (экспорт): <https://neo4j.com/docs/labs/apoc/current/export/json/>
4. Дкоументация Apoc (импорт): <https://neo4j.com/docs/labs/apoc/current/import/load-json/>

### ПРИЛОЖЕНИЕ А. ДОКУМЕНТАЦИЯ ПО СБОРКЕ И РАЗВЕРТЫВАНИЮ ПРИЛОЖЕНИЯ

Для использования приложения требуется наличие docker.

Установка приложения выполняется посредством команды:

docker pull trasil/marksdb

Запуск приложения выполняется с помощью команды:

Docker run –p 3000:3000 trasil/marksdb

После запуска приложение становится доступно по адресу:

http://localhost:3000/