

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: 16 ИС учета аспирантских достижений (Neo4j)

Студенты гр. 7382

Филиппов И.С.

Бахеров Д.В.

Дрозд А.С.

Преподаватель

Заславский М.М.

Санкт-Петербург

2020

ЗАДАНИЕ

НА ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студенты Филиппов И.С., Бахеров Д.В., Дрозд А.С.

Группа 7382

Тема работы: ИС учета аспирантских достижений (Neo4j)

Исходные данные:

Реализовать информационную систему учёта подготовки аспирантов к защите кандидатской работы

Содержание пояснительной записки:

«Содержание», «Введение», «Качественные требования к решению»,
«Сценарий использования», «Модель данных», «Разработка приложения»,
«Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 10 страниц.

Дата выдачи задания: 17.09.2020

Дата сдачи реферата: 01.02.2021

Дата защиты реферата: 01.02.2021

Студенты гр. 7382

Филиппов И.С.

Бахеров Д.В.

Дрозд А.С.

Преподаватель

Заславский М.М.

СОДЕРЖАНИЕ

Введение

1. Качественные требования к решению
2. Сценарии использования
 - 2.1. Макет пользовательского интерфейса
 - 2.2. Описание возможных сценариев использования
3. Модель данных
 - 3.1. Нереляционная модель данных
 - 3.2. Аналог модели данных для реляционной БД
4. Разработанное приложение

Заключение

Список использованных источников

ВВЕДЕНИЕ

Реализовать веб-приложение для сбора данных о подготовке аспирантов к защите кандидатской работы через выполнение промежуточных заданий.

Работа с данными написана на языке Python с использованием пакета Neo4j. Графический интерфейс написан на Python с использованием пакета Flask.

1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

Веб-приложение с некоторым UI, который позволяет:

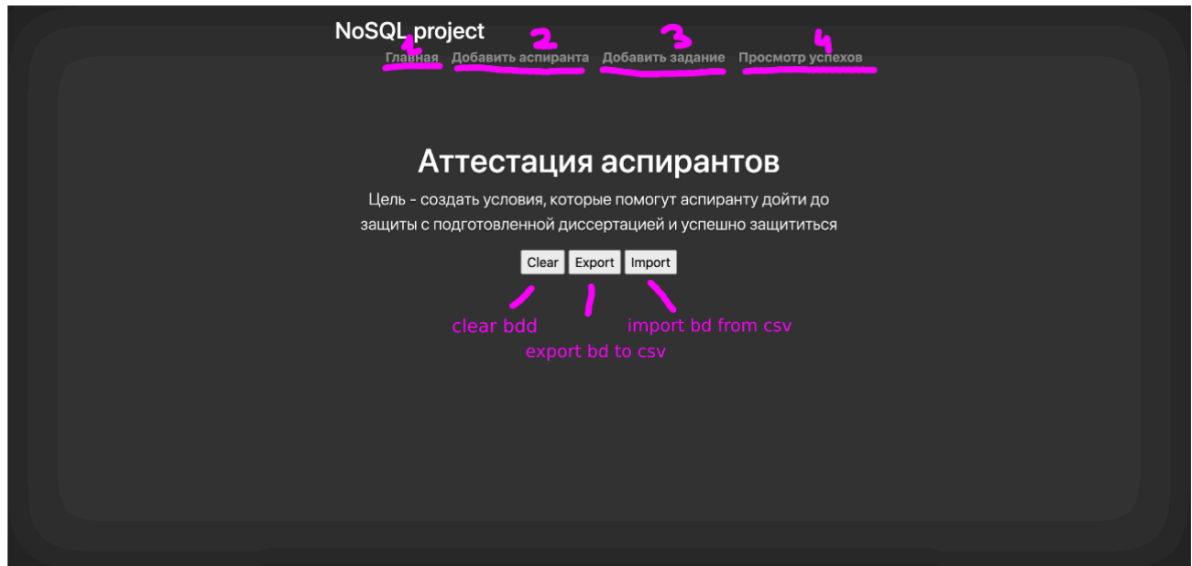
- Добавлять аспирантов
- Добавлять выполненные работы
- Осуществлять поиск и просмотр выполненных конкретным аспирантом работ

Использована Neo4j в качестве СУБД

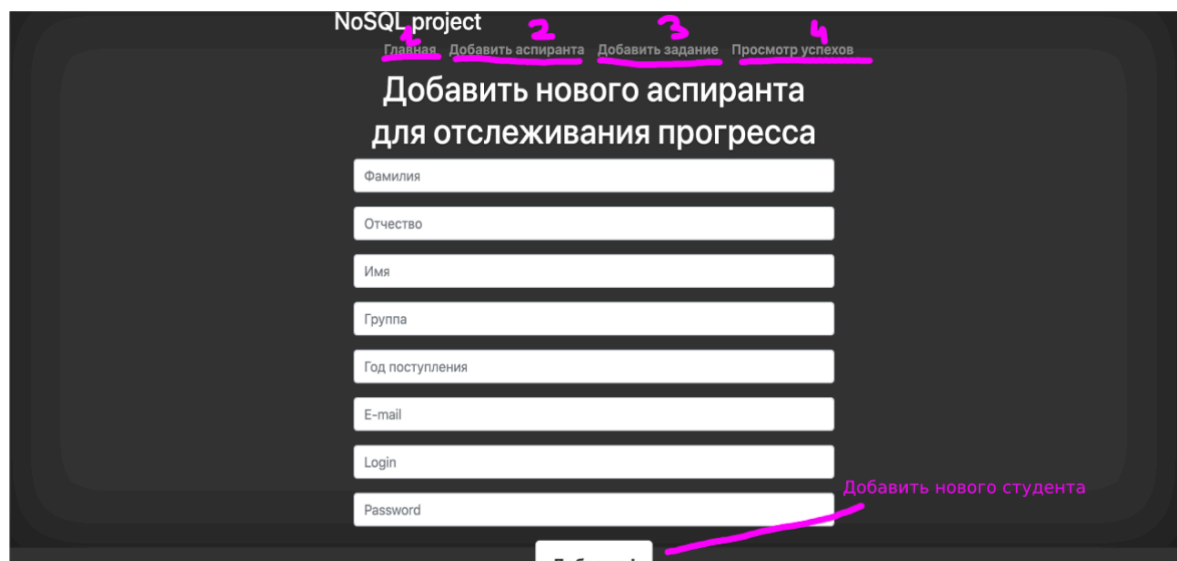
2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

2.1. Макет пользовательского интерфейса

1. Домашняя страница: очистка СУБД, импорт и экспорт данных из СУБД



2. Добавление нового аспиранта



3. Добавление новой работы:

NoSQL project

1 Главная 2 Добавить аспиранта 3 Добавить задание 4 Просмотр успехов

Добавить выполненное задание

Индекс

Семест

Ссылка на документ

Login

Password

Добавить!

Добавить задание, если есть студент с таким логином-паролем

4. Просмотр статистики конкретного студента:

NoSQL project

1 Главная 2 Добавить аспиранта 3 Добавить задание 4 Просмотр успехов

Просмотр успехов

Фамилия

Имя

Отчество

Посмотреть!

Посмотреть список со статистикой

1. Work 0. Type: Опубликованная статья. Semester: 2. Link: tmp@mail.ru.	Works list
2. Work 1. Type: Обзор литературы. Semester: 1. Link: tmp@mail.ru.	
Оценка на данный момент: 2	
Mark	

1. Type: Опубликованная статья. Semester: 7. Link: tmp@mail.ru.
2. Type: Опубликованная статья. Semester: 6. Link: tmp@mail.ru.
3. Type: Опубликованная статья. Semester: 5. Link: tmp@mail.ru.
4. Type: Опубликованная статья. Semester: 4. Link: tmp@mail.ru.
5. Type: Опубликованная статья. Semester: 3. Link: tmp@mail.ru.
6. Type: Опубликованная статья. Semester: 2. Link: tmp@mail.ru.
7. Type: Обзор литературы. Semester: 1. Link: tmp@mail.ru.

Оценка на данный момент: 3

Mark

Works list

2.2 Описание возможных сценариев использования.

1. Добавление нового аспиранта. Для этого вводятся информационные поля (от имени и заканчивая email), и логин и пароль -- они нужны для добавления выполненных работ для отслеживания прогресса.

2. Добавление новой работы. Указание её типа (по старшинству из таблицы работ для получения оценки аспирантом), семестра, в котором она была сдана и ссылки на google-disc для просмотра и редактирования. Для того, чтобы добавить работу конкретному студенту, необходимо указать его логин и пароль -- в противном случае работа не добавится.

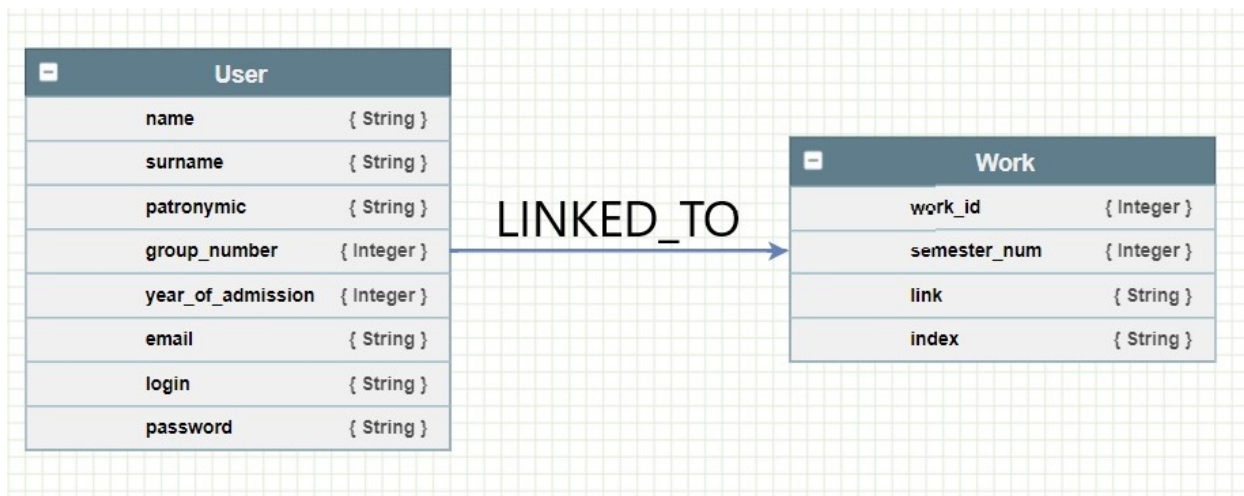
3. Поиск конкретного аспиранта -- необходимо указать его Имя-Фамилию-Отчество, и будет выведен список выполненных работ. Для того, чтобы посмотреть достижения какого-либо аспиранта не нужно знать его логин-пароль.

4. Просмотр найденных работ конкретного аспиранта в виде списка для отслеживания его прогресса и просмотр достигнутой на данный момент оценки.

3.

МОДЕЛЬ ДАННЫХ

Схема базы данных Neo4j



Список сущностей модели

User, Work

Описание назначений коллекций, типов данных и сущностей

Коллекции:

Название	Описание
User	Содержит информацию о пользователе
Work	Содержит информацию о работе

Описание User:

Название	Тип данных	Описание
name	String	Имя пользователя
surname	String	Фамилия пользователя

patronymic	String	Отчество пользователя
group_number	Integer	Номер группы
year_of_admission	Integer	Год поступления
email	String	Электронная почта
login	String	Логин
password	String	Пароль

Описание Work:

Название	Тип данных	Описание
work_id	Integer	Идентификационный номер работы
semester_num	Integer	Семестр, в течении которого была выполнена данная работа
link	String	Ссылка на документ
index	String	Показатель работы

Удельный объем информации. Оценка удельного объема информации, хранимой в модели NoSql

Тип данных	Размер
String	2 байта/символ
Integer	4 байта

Размер одной User: $User_size = 2*name_length + 2*surname_length + 2*patronymic_length + 4 + 4 + 2*email_length + 2*login_length + 2*password_length = 8 + 2*(name_length + surname_length + patronymic_length + email_length + login_length + password_length)$

Размер одной Work: $Work_size = 4 + 4 + 2*link_length + 2*index_length = 8 + 2*(link_length + index_length)$

Общий объем БД:

$USERS_NUM * User_size + WORKS_NUM * Work_size$

USERS_NUM - количество пользователей

WORKS_NUM - количество всех работ всех пользователей

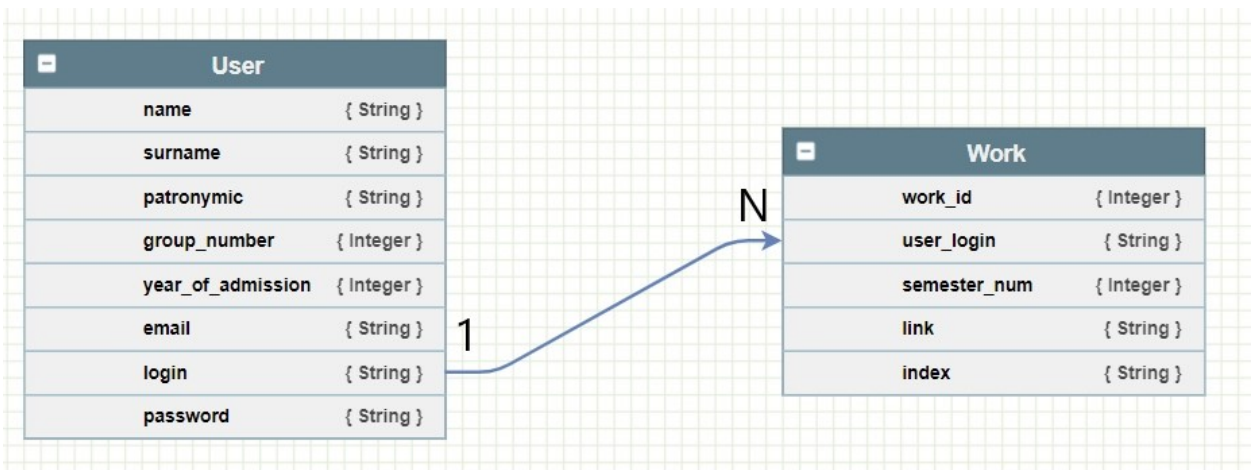
Избыточность модели

Отсутствует избыточность данных. Это означает, что существующие данные в базе данных не могут быть образованы путем объединения или комбинирования прочих данных той же базы данных.

Направление роста модели при увеличении количества объектов каждой сущности.

При росте количества сущностей User сущности Work автоматически не добавляются. Увеличение количества сущностей Work не влияет на количество Users, добавление работы подразумевает лишь одного пользователя.

Схема базы данных Sql



Удельный объем информации. Оценка удельного объема информации, хранимой в модели Sql

Размер одной User: $User_size = 2 * name_length + 2 * surname_length + 2 * patronymic_length + 4 + 4 + 2 * email_length + 2 * login_length + 2 * password_length = 8 + 2 * (name_length + surname_length + patronymic_length + email_length + login_length + password_length)$

Размер одной Work: $Work_size = 4 + 4 + 2 * link_length + 2 * index_length + 2 * user_login_length = 8 + 2 * (link_length + index_length + user_login_length)$

Общий объем БД:

$USERS_NUM * User_size + WORKS_NUM * Work_size$

USERS_NUM - количество пользователей

WORKS_NUM - количество всех работ всех пользователей

Количество задействованных коллекций:

User, Work

Пример хранения данных в БД NoSql

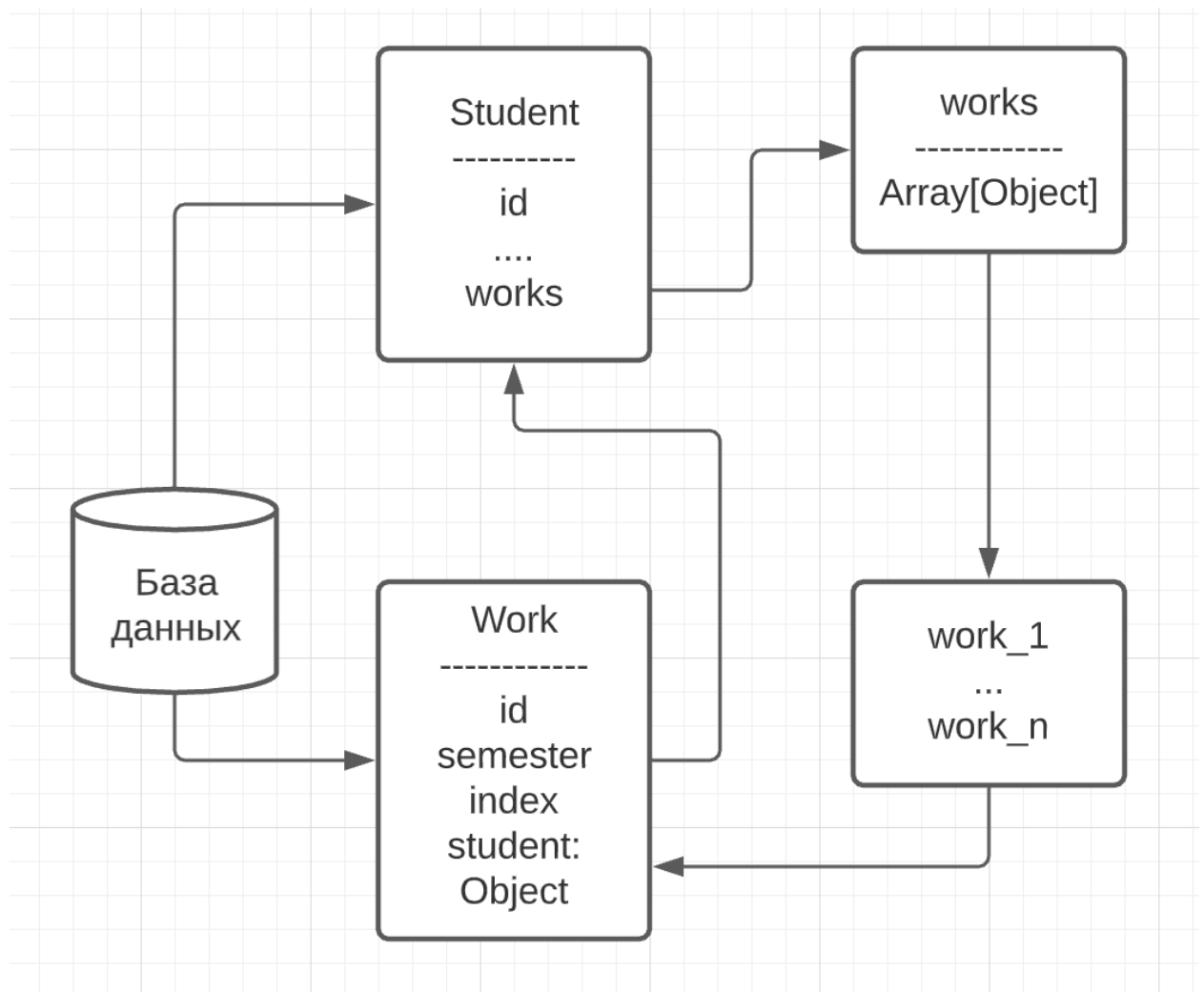


\$ MATCH (u:User {name: "Daniil"})-[:LINKED_TO]->(w) RETURN u, w		
	u	w
Graph		
Table	1	
Text		
Code		
	<pre>{ "identity": 0, "labels": ["User"], "properties": { "patronymic": "Vladislavovich", "password": "input_password", "group_number": 7382, "year_of_admission": 2017, "surname": "Baherov", "name": "Daniil", "login": "input_login", "email": "input_email" } }</pre>	<pre>{ "identity": 40, "labels": ["Work"], "properties": { "link": "link", "work_id": 1, "index": "Conference_report", "semester_num": 6 } }</pre>

Сравнение SQL и NoSQL

SQL реализация данной БД занимает больше памяти, так как в сущности Work нужно дополнительно хранить поле `user_login` для связи с родительской сущностью User, в результате чего происходит дублирование информации.

Схема базы данных MongoDB



Модель состоит из 2 коллекций:

Graduate - хранит информацию о студенте

- * id - Int - уникальный идентификатор человека
- * name - String - имя
- * surname - String - фамилия
- * patronymic - String - отчество
- * group number - Int - номер группы
- * year of admission - Int - год поступления
- * email - String - email
- * login - String - логин, чтобы добавлять себе сделанные работы
- * password - String - пароль, чтобы добавлять себе сделанные работы)
- * works - Object - работы, если есть

* p_id - Array[Object] - список работ

Work - хранит информацию о работе

* id - int - уникальный идентификатор работы

* semester - Int - семестр, в котором она была выполнена

* index - Int - номер работы в списке работ аспиранта

* link - String - ссылка на google doc работы

* student - Object - студент, сделавший её

* f_id - Int - идентификатор студента

4.

РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

Добавление аспиранта

Главная Добавить аспиранта Добавить задание Просмотр успехов

Добавить нового аспиранта для отслеживания прогресса

Филиппов

Сергеевич

Игорь

7382

2016

filippovis.igor@yandex.ru

123

123|

Добавить!

Добавление работы

Добавить выполненное задание

Индекс

Семест

Ссылка на документ

Login

Password

Добавить!

NoSQL project

Главная Добавить аспиранта Добавить задание Просмотр успехов

Добавить выполненное задание

2

6

tmp2@mail.ru

123

123|

Добавить!

Просмотр статистики

NoSQL project

[Главная](#) [Добавить аспиранта](#) [Добавить задание](#) [Просмотр успехов](#)

Просмотр успехов

Фамилия

Имя

Отчество

Посмотреть!

NoSQL project

[Главная](#) [Добавить аспиранта](#) [Добавить задание](#) [Просмотр успехов](#)

Просмотр успехов

Филиппов

Игорь

Сергеевич

Посмотреть!

Work 0. Type: Грант / конкурс / программа для ЭВМ / методичка. Semester: 6. Link: tmp2@mail.ru.

Work 1. Type: Опубликованная статья. Semester: 2. Link: tmp@mail.ru.

ЗАКЛЮЧЕНИЕ

В ходе работы было получено веб-приложение, с помощью которого (после доработки фронтенда), можно осуществлять учёт достижений аспирантов при подготовке к защите.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. <https://neo4j.com/docs>
2. <https://flask-doc.readthedocs.io/en/latest/>
3. <https://docs.docker.com/>
4. <https://aws.amazon.com/ru/nosql/>
5. <https://docs.mongodb.com/>