

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: Каталог учебных планов и программ

Студентка гр. 0383	_____	Александрович В.П.
Студентка гр. 0383	_____	Куртова К.А.
Студент гр. 0383	_____	Парфенов В.М.
Преподаватель	_____	Заславский М. М.

Санкт-Петербург
2023

ЗАДАНИЕ

Студенты

Александрович В.П.

Куртова К.А.

Парфенов В.М.

Группа 0383

Тема проекта: Каталог учебных планов и программ

Исходные данные:

Необходимо реализовать сервис для управления и визуализацией (в виде графа) учебных программ и планов в рамках ВУЗа. Пользователи - преподаватели, студенты. Необходимые (но не достаточные) требования - страница учебной программы, учебной дисциплины, анализ учебных планов (общие дисциплины, отличия, сложность).

Содержание пояснительной записки:

«Содержание»

«Введение»

«Качественные требования к решению»

«Модель данных»

«Разработка приложения»

«Вывод»

«Приложение»

Предполагаемый объем пояснительной записки:

Не менее 10 страниц.

Дата выдачи задания: 01.09.2023

Дата сдачи реферата: 25.12.2023

Дата защиты реферата: 25.12.2023

Студентка		Александрович В.П.
Студентка		Куртова К.А.
Студентка		Парфенов В.М.
Преподаватель		Заславский М. М.

АННОТАЦИЯ

В рамках данного курса предполагалось разработать в команде веб-приложение на одну из поставленных тем. Была выбрана тема создания электронного каталога учебных планов и программ с использованием графовой СУБД Neo4j. Пользователь приложения должен обладать возможностью редактирования учебных программ и планов в рамках ВУЗа.

Найти исходный код и всю дополнительную информацию можно по ссылке: <https://github.com/moevm/nosql2h23-study-progs.git>

SUMMARY

As part of this course, the team was planned to develop a web application on one of the assigned topics. The chosen topic is as following: creating an electronic catalog of curricula and programs using the Neo4j graph DBMS. The application user must have the ability to edit curricula and plans of the university.

You can find the source code and all additional information at the link: <https://github.com/moevm/nosql2h23-study-progs.git>

СОДЕРЖАНИЕ

1.	Введение	6
2.	Качественные требования к решению	6
3.	Сценарии использования	6
4.	Модель данных	11
5.	Разработанное приложение	25
6.	Выводы	28
7.	Приложения	29
8.	Используемая литература	30

1. ВВЕДЕНИЕ

Цель работы – создать высокопроизводительное и удобное веб-приложение для хранения информации об учебных планах и программах университета «ЛЭТИ». Для решения поставленной цели было решено хранить и запрашивать информацию, а также производить над ней изменения с помощью графовой СУБД Neo4j.

2. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

Все виды манипуляции с информацией, такие как фильтрация, должны производиться через запросы базе данных. Пользователю должны быть доступны следующие возможности: просмотр страниц учебной программы, учебной дисциплины, анализ учебных планов (общие дисциплины, отличия, сложность), просмотр визуализации базы данных в виде графа, добавление учебных программ, учебных планов.

3. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

Макеты UI

1. Экран входа в приложение (Рис. 1).

Каталог учебных планов и программ ЛЭТИ / Вход

Логин

Пароль

Войти

Регистрация

Рисунок 1 – страница входа

2. Страница списка образовательных программ (Рис. 2).

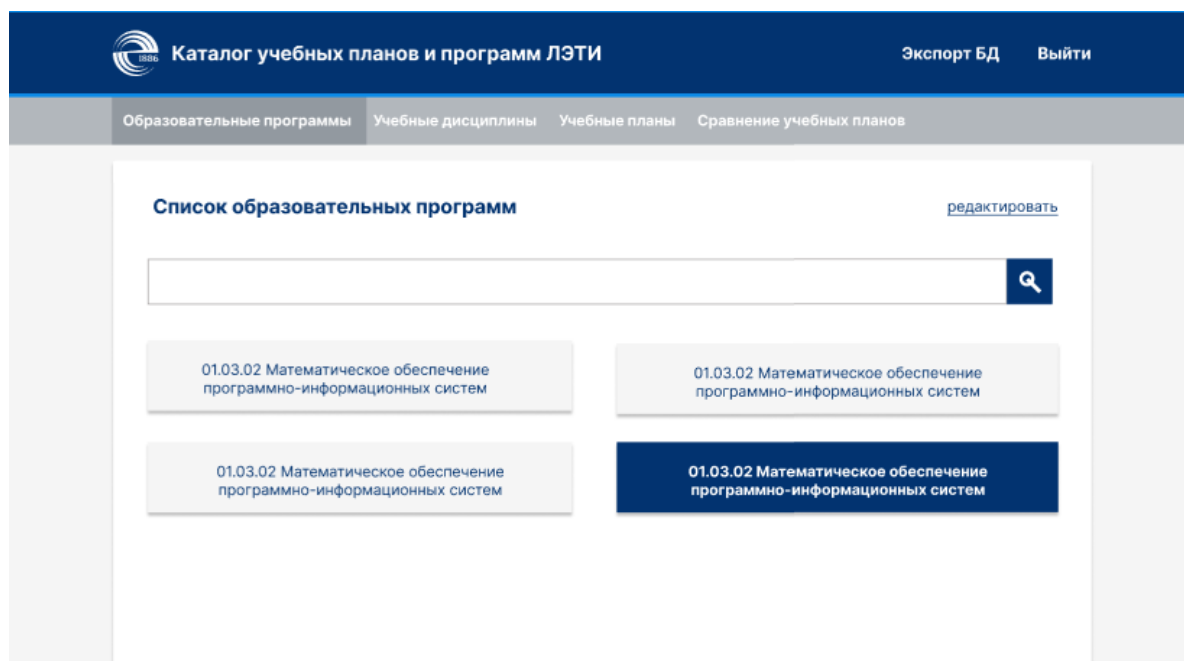


Рисунок 2 – страница списка образовательных программ

3. Страница рабочей программы дисциплины (Рис. 3).

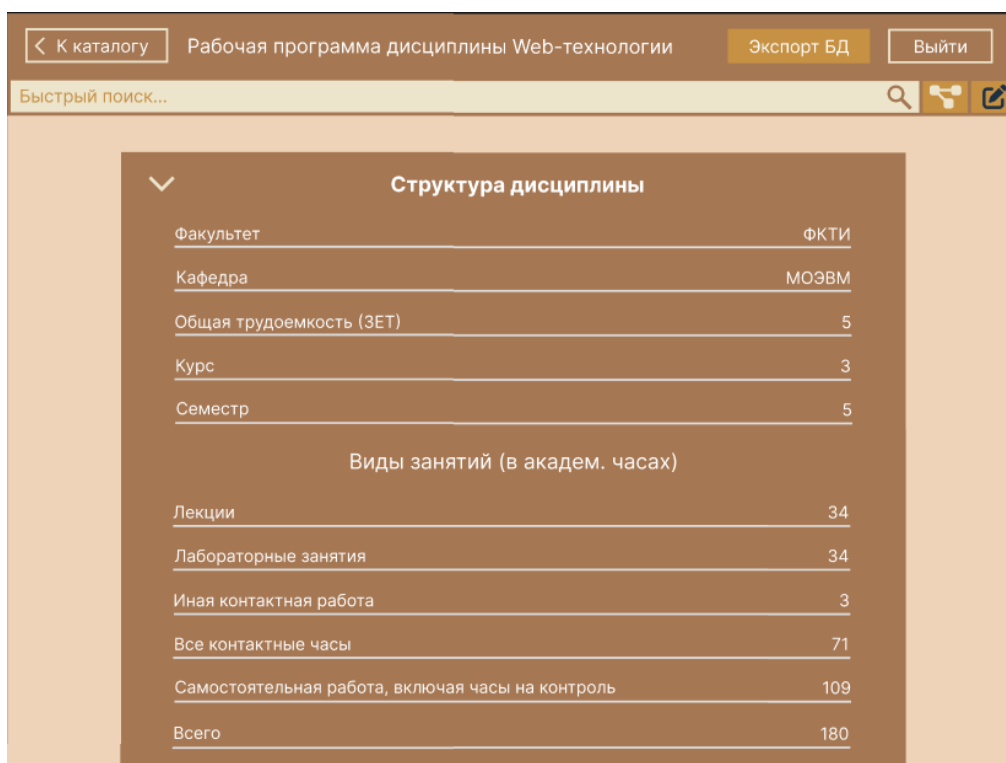


Рисунок 3 – страница рабочей программы дисциплины

4. Страница учебного плана (Рис. 4).

Общая информация

	Курс 1			Курс 2			Курс 3			Курс 4			Итого
	Сем. 1	Сем. 2	Всего	Сем. 3	Сем. 4	Всего	Сем. 5	Сем. 6	Всего	Сем. 7	Сем. 8	Всего	
Теоретическое обучение и практики	17	17	34	17	17	34	17	17	34	17	8	25	127
Э Экзаменационные сессии	3	3	6	3	2	5	3	2	5	3		3	19
У Учебная практика					2	2							2
П Производственная практика								2	2				2
Пд Преддипломная практика										6	6	6	6
Д Подготовка к процедуре защиты и защита выпускной квалификационной работы										6	6	6	6
К Каникулы	1 1/6	8 3/6	9 4/6	1 1/6	7 3/6	8 4/6	1 1/6	7 3/6	8 4/6	1 1/6	8 3/6	9 4/6	36 4/6
* Нерабочие праздничные дни (не включая воскресенье)	1 2/6 (8 дн)	1 (6 дн)	2 2/6 (14 дн)	1 2/6 (8 дн)	1 (6 дн)	2 2/6 (14 дн)	1 2/6 (8 дн)	1 (6 дн)	2 2/6 (14 дн)	1 2/6 (8 дн)	1 (6 дн)	2 2/6 (14 дн)	9 2/6 (56 дн)
Продолжительность обучения	более 39 нед.			более 39 нед.			более 39 нед.			более 39 нед.			
Итого	22 3/6	29 3/6	52	22 3/6	29 3/6	52	22 3/6	29 3/6	52	22 3/6	29 3/6	52	208

Сравнение учебных планов программы

Учитывать планы других дисциплин ☐

01.03.02 № 338-21

01.03.02 № 338-22

01.03.02 № 338-23

Анализ

Рисунок 4 – страница учебного плана

5. Страница сравнения учебных планов (Рис. 5).

Экспорт БД Выйти

Сравнение нагрузки в дисциплинах

Параметры сравнения

☒ Количество часов лекций
 ☐ Количество часов на самостоятельную работу

☐ Количество часов на лабораторные работы
 ☐ Количество часов на итоговую контрольную работу

☐ Количество часов на практические работы
 ☐ Количество часов всего

☐ По семестрам

Введите название дисциплины

Дисциплина	01.03.02 № 338-20	01.03.02 № 338-21
Философия	56	40

Рисунок 5 – страница сравнения учебных планов

Описание сценариев использования

Просмотр каталога учебных планов и программ

1. Пользователь находится на странице авторизации.
2. Пользователь вводит логин и пароль.
3. Пользователь нажимает на кнопку "Вход".
4. Пользователь попадает на страницу просмотра каталога учебных планов и программ.

Просмотр страницы дисциплины

1. Пользователь находится на странице просмотра каталога учебных планов и программ.
2. Пользователь переходит на вкладку «Учебные дисциплины».
3. Пользователь выбирает дисциплину из списка.
4. Пользователь попадает на страницу просмотра программы дисциплины.

Просмотр страницы образовательной программы

1. Пользователь находится на странице просмотра каталога учебных планов и программ.
2. Пользователь переходит на вкладку «Образовательные программы».
3. Пользователь выбирает программу из списка.
4. Пользователь попадает на страницу просмотра образовательной программы.

Просмотр страницы учебного плана

1. Пользователь находится на странице просмотра каталога учебных планов и программ.
2. Пользователь переходит на вкладку «Учебные планы».
3. Пользователь выбирает план из списка.

4. Пользователь попадает на страницу просмотра учебного плана.

Сравнительный анализ учебных планов

1. Пользователь находится на любой странице приложения.
2. Пользователь нажимает на вкладку «Сравнение учебных планов».
3. Пользователь выбирает два плана из списка "Сравнение учебных планов".
4. Пользователь нажимает на кнопку "Анализ".
5. Пользователь попадает на страницу сравнения учебных планов.

Просмотр графа дисциплин, программ и планов

1. Пользователь находится на любой странице приложения.
2. Пользователь нажимает на вкладку «Визуализация графа».
3. Пользователь переходит на страницу просмотра графа дисциплин, программ и планов.

Добавление страниц

1. Пользователь находится на странице каталога учебных планов, образовательной программы, учебного плана или программы дисциплины.
2. Пользователь нажимает на кнопку добавления страницы над списком.
3. Пользователь заполняет поля информации на странице добавления.
4. Пользователь добавляет страницу с помощью кнопки "Добавить".

Экспорт данных

1. Пользователь нажимает на кнопку "Экспортировать БД".
2. Начинается загрузка базы данных на компьютер пользователя.

4. МОДЕЛЬ ДАННЫХ

Нереляционная модель данных

Графическое представление модели (Рис. 6).

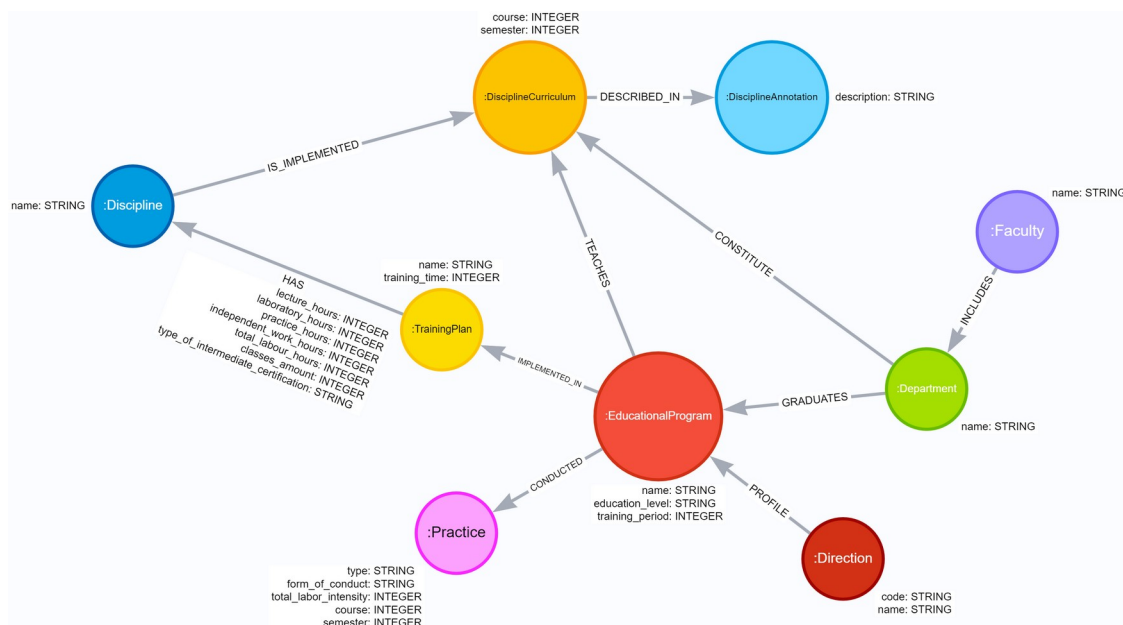


Рисунок 6 - графическое представление модели Neo4j

Описание узлов

- Faculty - Факультет
 - name - название факультета. Тип STRING.
- Department - Кафедра
 - name - название кафедры. Тип STRING.
- Direction - Направление подготовки
 - name - название направления подготовки. Тип STRING.
 - code - код направления подготовки. Тип STRING.
- EducationalProgram - Образовательная программа
 - name - название образовательной программы. Тип STRING.
 - educational_level - уровень образования. Тип STRING.
 - training_period - срок обучения. Тип INTEGER.
- Practice - Программа практики

- type - вид практики (учебная, производственная, преддипломная).
- form_of_conduct - форма проведения. Тип STRING.
- total_labor_intensity - общая трудоёмкость (ЗЕТ). Тип INTEGER.
- course - курс. Тип INTEGER.
- semester - семестр. Тип INTEGER.
- TrainingPlan - Учебный план программы
 - name - название учебного плана. Тип STRING.
 - training_time - продолжительность обучения. Тип INTEGER.
- Discipline - Дисциплина (общие сведения)
 - name - название дисциплины. Тип STRING.
- DisciplineCurriculum - Учебная программа дисциплины, различная для каждого направления
 - course - курс. Тип INTEGER.
 - semester - семестр. Тип INTEGER.
- DisciplineAnnotation - Аннотация дисциплины, различная для каждого направления
 - description - описание дисциплины. Тип STRING.

Описание связей

- (:Faculty)-[:INCLUDES]->(:Department) - Принадлежность кафедры определённому факультету
- (:Department)-[:GRADUATES]->(:EducationalProgram) - У каждой образовательной программы есть своя выпускающая кафедра
- (:EducationalProgram)-[:CONDUCTED]->(:Practice) - Программа практики относится к определённой образовательной программе
- (:Direction)-[:PROFILE]->(:EducationalProgram) - Специальность/направление подразделяется на несколько образовательных программ
- (:TrainingPlan)-[:HAS]->(:Discipline) - Дисциплина включена в учебный план
 - lecture_hours - лекции (академ. часов).

- laboratory_hours - лабораторные занятия (академ. часов).
- practice_hours - практические занятия (академ. часов).
- independent_work_hours - самостоятельная работа (академ. часов).
- total_labour_intensity - общая трудоёмкость (ЗЕТ).
- classes_amount - кол-во занятий.
- type_of_intermediate_certification - вид промежуточной аттестации.
- (:Discipline)-[:IMPLEMENTED_IN]->(:DisciplineCurriculum) - Рабочие программы дисциплин с одним названием по-разному реализованы на разных образовательных программах
- (:EducationalProgram)-[:TEACHES]->(:DisciplineCurriculum) - Рабочая программа дисциплины для данного направления
- (:Department)-[:CONSTITUTE]->(:DisciplineCurriculum) - Кафедра утверждает программу рабочей дисциплины
- (:DisciplineCurriculum)-[:DESCRIBED_IN]->(:DisciplineAnnotation) - Каждая дисциплина имеет своё уникальное описание

Оценка удельного объема информации, хранимой в модели

Таблица 1 – оценка удельного объема информации

Тип	Размер, байт	Комментарий
INTEGER	4	
STRING	100/1000	В среднем присутствуют строки до 100 символов, однако описания дисциплин намного длиннее (около 1000 символов)
LONG	8	Используется для id узлов и связей

В Neo4j связи представлены парой id узлов и собственным id. Таким образом, объем одной связи без атрибутов - 24 байт. Все связи однонаправленные.

Каждый узел имеет свой уникальный id, к каждому рассчитанному объёму со всеми атрибутами прибавляется 8 байт.

Выразим объем базы данных через количество образовательных программ (Educational Program) *E*.

Таблица 2 – расчет объема узлов

Таблица	Объём одной строки, байт	Среднее количество	Общий объём, байт	Комментарий
Educational Program	212	E	212 E	-
Training Plan	112	3 E	336 E	Каждая образовательная программа имеет в среднем 3 учебных плана на разные года приёма
Discipline Curriculum	16	30 E	480 E	Каждая программа в среднем обучает 30 дисциплинам
Practice	220	3 E	660 E	Каждая образовательная программа имеет в среднем 3 документа с описанием практик (учебная, производственная, преддипломная)
Direction	208	1/4 E	52 E	Каждое направление в среднем имеет три образовательных программы
Discipline	108	3 E	324 E	Каждая программа имеет в среднем 10 учебных программ
Discipline Annotation	1008	30 E	30240 E	Описаний дисциплин столько же, сколько и реализаций
Department	108	1/4 E	52 E	Каждая кафедра в среднем реализует 4 образовательных программы
Сумма	-	-	32356 E	-

Таблица 3 – расчет объема связей

Связь	Объём одной	Среднее	Общий	Комментарий
-------	-------------	---------	-------	-------------

	сущности, байт	количество	объём, байт	
(:Faculty)- [:INCLUDES]- >(:Department)	24	1/4 E	6 E	Связей столько же, сколько кафедр
(:Department)- [:GRADUATES]- >(:EducationalProgram)	24	E	24 E	Связей столько же, сколько образовательных программ
(:EducationalProgram)- [:CONDUCTED]- >(:Practice)	24	3 E	72 E	Связей столько же, сколько документов практик
(:Direction)- [:PROFILE]- >(:EducationalProgram)	24	E	24 E	Связей столько же, сколько образовательных программ
(:TrainingPlan)-[:HAS]- >(:Discipline)	132	150 E	19800 E	Каждый из учебных планов программы содержит ~50 дисциплин
(:Discipline)- [:IMPLEMENTED_IN]- >(:DisciplineCurriculum)	24	30 E	720 E	Связей столько же, сколько реализаций программ дисциплин
(:EducationalProgram)- [:TEACHES]- >(:DisciplineCurriculum)	24	30 E	720 E	Связей столько же, сколько реализаций программ дисциплин
(:Department)- [:CONSTITUTE]- >(:DisciplineCurriculum)	24	30 E	720 E	Связей столько же, сколько реализаций программ дисциплин
(:DisciplineCurriculum)- [:DESCRIBED_IN]-	24	30 E	720 E	Связей столько же, сколько реализаций

>(:DisciplineAnnotation)				программ дисциплин
Сумма	-	-	22806 E	-

Общий объём базы данных: $32356 \text{ E} + 22806 \text{ E} = 55162 \text{ E}$

В данный момент значения равны: $E \sim 240$

Текущий объём базы данных: $\sim 12,9 \text{ Mb}$

Избыточность модели

Избыточными данными в данной модели можно считать связи $(\text{:EducationalProgram})\text{--}[\text{:TEACHES}]\text{--}>(\text{:DisciplineCurriculum})$ и $(\text{:Department})\text{--}[\text{:CONSTITUTE}]\text{--}>(\text{:DisciplineCurriculum})$, так как DisciplineCurriculum так или иначе связан с EducationalProgram и Department через TrainingPlan. Данные связи нужны лишь для удобства использования БД. Поэтому избыточность модели близка к 1.

Направление роста модели

Наибольшее влияние на размер БД при росте количества сущностей будут оказывать сущности, связанные с дисциплинами, а именно Discipline (в основном из-за связи между TrainingPlan и Discipline), DisciplineCurriculum и, соответственно, DisciplineAnnotation. Каждый из учебных планов имеет свой уникальный DisciplineCurriculum для каждой из описанных на нем дисциплин, которые, помимо связи с EducationalProgram, также связаны с кафедрой, которая их утверждает, и каждая из дисциплин соответственно имеет своё уникальное и довольно затратное по объёму (в среднем до 1000 байт) описание. С каждой новой сущностью типа DisciplineCurriculum к объёму БД прибавляется в среднем 1120b.

Примеры запросов

- Получить список образовательных программ
`MATCH (n:EducationalProgram) RETURN n LIMIT 25;`
- Получить список учебных планов
`MATCH (n:TrainingPlan) RETURN n LIMIT 25;`
- Получить список рабочих программ
`MATCH (n:DisciplineCurriculum) RETURN n LIMIT 25;`
- Получить список дисциплин, которые есть и первом учебном плане, и во втором
`MATCH(:TrainingPlan {name: "Учебный план № 338-23 (прием 2023)"})-[:HAS]-(d:Discipline)-[:HAS]-(:TrainingPlan {name: "Учебный план № 338-22 (прием 2022)"})
RETURN d`
- Получить список дисциплин, которые присутствуют в одном учебном плане и отсутствуют в другом учебном плане
`MATCH ((:TrainingPlan {name: "Учебный план № 338-22 (прием 2022)"})-[:HAS]-(a:Discipline) WHERE not (a)-[:HAS]-(:TrainingPlan {name: "Учебный план № 338-23 (прием 2023)"}))
RETURN a;`
- Получить список дисциплин на конкретной кафедре
`MATCH(:Department {name: 'МО ЭВМ'})-[:CONSTITUTE]-(
:DisciplineCurriculum)-[:IS_IMPLEMENTED]-(d:Discipline)
RETURN d;`

Аналог модели данных для SQL СУБД

Графическое представление модели (Рис. 8.)

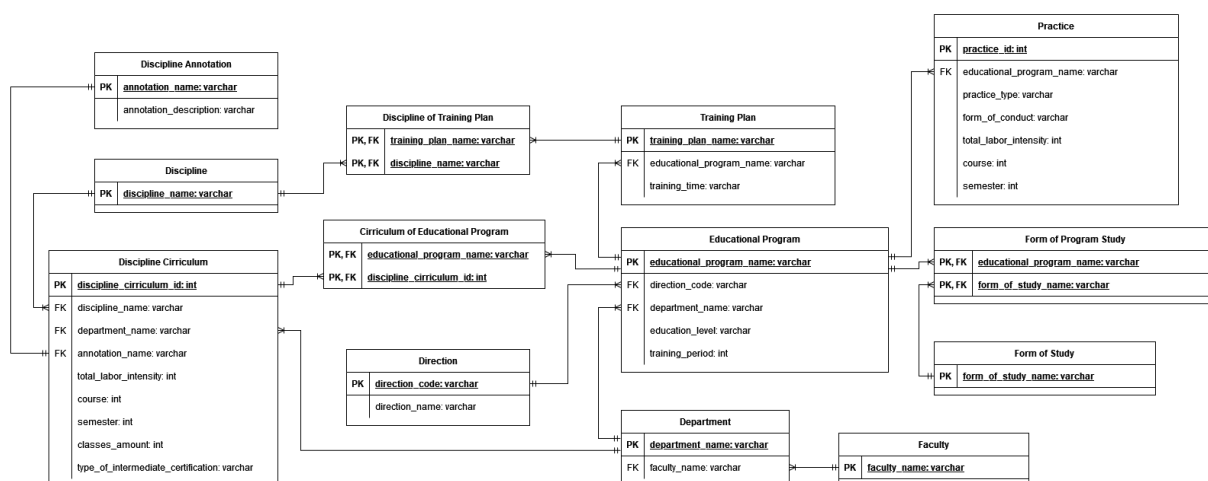


Рисунок 7 – графическое представление модели

- Таблица “Discipline” - Дисциплина
 - **Ключ:** discipline_name - уникальное название дисциплины. Тип varchar. V = 100b
- Таблица “Discipline Annotation” - Аннотация дисциплины
 - **Ключ:** annotation_name - уникальное название аннотации дисциплины. Тип varchar. V = 100b
 - annotation_description - текст аннотации. Тип varchar. V = 1000b
- Таблица “Discipline Curriculum” - Учебная программа дисциплины
 - **Ключ:** discipline_curriculum_id - id уч. программы дисциплины. Тип int. V = 4b
 - **Внешний ключ:** discipline_name - название дисциплины. Тип varchar. V = 100b
 - **Внешний ключ:** department_name - кафедра. Тип varchar. V = 100b
 - **Внешний ключ:** annotation_name - уникальное название аннотации дисциплины. Тип varchar. V = 100b
 - total_labor_intensity - общая трудоёмкость (ЗЕТ). Тип int. V = 4b
 - course - курс. Тип int. V = 4b
 - semester - семестр. Тип int. V = 4b
 - classes_amount - кол-во занятий. Тип int. V = 4b

- type_of_intermediate_certification - вид промежуточной аттестации.
Тип varchar. V = 100b
- Таблица “Discipline of Training Plan” - Образовательная программа/Учебный план
 - **Ключ, внешний ключ:** training_plan_name - название учебного плана. Тип varchar. V = 100b
 - **Ключ, внешний ключ:** discipline_name - название дисциплины.
Тип int. V = 4b
- Таблица “Training Plan” - Учебный план
 - **Ключ:** training_plan_name - название учебного плана. Тип varchar. V = 100b
 - **Внешний ключ:** educational_program_name - образовательная программа. Тип varchar. V = 100b
 - training_time - продолжительность обучения. Тип varchar. V = 100b
- Таблица “Educational Program” - Образовательная программа
 - **Ключ:** educational_program_name - название образовательной программы. Тип varchar. V = 100b
 - **Внешний ключ:** direction_code - код направления. Тип varchar. V = 100b
 - **Внешний ключ:** department_name - кафедра. Тип varchar. V = 100b
 - education_level - уровень образования. Тип varchar. V = 100b
 - training_period - срок обучения. Тип int. V = 4b
- Таблица “Curriculum of Educational Program” - Учебная программа дисциплины/Образовательная программа
 - **Ключ, внешний ключ:** educational_program_name - название образовательной программы. Тип varchar. V = 100b
 - **Ключ, внешний ключ:** discipline_curriculum_id - id уч. программы дисциплины. Тип int. V = 4b
- Таблица “Direction” - Направление
 - **Ключ:** direction_code - код направления. Тип varchar. V = 100b

- direction_name - название направления. Тип varchar. V = 100b
- Таблица “Department” - Кафедра
 - **Ключ:** department_name - название кафедры. Тип varchar. V = 100b
 - **Внешний ключ:** faculty_name - факультет. Тип varchar. V = 100b
- Таблица “Faculty” - Факультет
 - **Ключ:** faculty_name - название факультета. Тип varchar. V = 100b
- Таблица “Practice” - Практика
 - **Ключ:** practice_id - id практики. Тип int. V = 4b
 - **Внешний ключ:** educational_program_name - название образовательной программы. Тип varchar. V = 100b
 - practice_type - вид практики. Тип varchar. V = 100b
 - form_of_conduct - форма проведения. Тип varchar. V = 100b
 - total_labor_intensity - общая трудоёмкость (ЗЕТ). Тип int. V = 4b
 - course - курс. Тип int. V = 4b
 - semester - семестр. Тип int. V = 4b
- Таблица “Form of Study” - Форма обучения
 - **Ключ:** form_of_study_name - название формы обучения. Тип varchar. V = 100b
- Таблица “Form of Program Study” - Форма обучения программы
 - **Ключ, внешний ключ:** educational_program_name - название образовательной программы. Тип varchar. V = 100b
 - **Ключ, внешний ключ:** form_of_study_name - название формы обучения. Тип varchar. V = 100b

Оценка удельного объёма информации

Тип	Размер, байт
varchar	100/1000
int	4

--	--

Средний размер одной строки таблицы (байт):

- Discipline: 100 байт.
- Discipline Annotation: $100 + 1000 = 1100$ байт.
- Discipline Curriculum: $4 + 100 + 100 + 100 + 4 + 4 + 4 + 4 + 100 = 420$ байт.
- Discipline of Training Plan: $100 + 4 = 104$ байт.
- Training Plan: $100 + 100 + 100 = 300$ байт.
- Educational Program: $100 + 100 + 100 + 100 + 4 = 404$ байт.
- Curriculum of Educational Program: $100 + 4 = 104$ байт.
- Direction: $100 + 100 = 200$ байт.
- Department: $100 + 100 = 200$ байт.
- Faculty: 100 байт.
- Practice: $4 + 100 + 100 + 100 + 4 + 4 + 4 = 316$ байт.
- Form of Study: 100 байт.
- Form of Program Study: $100 + 100 = 200$ байт.

Расчет объема базы данных.

Выразим объем базы данных через количество образовательных программ (Educational Program) E .

Таблица	Объем одной строки, байт	Среднее количество	Общий объем, байт	Комментарий
Educational Program	404	E	$404 E$	-
Training Plan	300	$3 E$	$900 E$	Каждая образовательная программа имеет в среднем 3 учебных плана на разные года приёма
Discipline Curriculum	420	$30 E$	$12600 E$	Каждая программа в среднем обучает 30 дисциплинам

Practice	316	3 E	948 E	Каждая образовательная программа имеет в среднем 3 документа с описанием практик (учебная, производственная, преддипломная)
Direction	200	1/4 E	50 E	Каждое направление в среднем имеет три образовательных программы
Discipline	100	3 E	300 E	Каждая программа имеет в среднем 10 учебных программ
Discipline Annotation	1100	30 E	33000 E	Описаний дисциплин столько же, сколько и реализаций
Discipline of Training Plan	104	150 E	15600 E	Каждый из 3 учебных планов программы содержит ~50 дисциплин
Curriculum of Educational Program	104	30 E	3120 E	Строк столько же, сколько реализаций программ дисциплин
Form of Program Study	200	E	200 E	Каждая образовательная программа в среднем реализует одну форму обучения
Department	200	1/4 E	50 E	Каждая кафедра в среднем реализует 4 образовательных программы
Faculty	100	10	1000	На сайте ЛЭТИ представлено ~10 факультетов (Объем пренебрежимо мал и не растет с увеличением БД)
Form of Study	100	3	300	Очная, Заочная, Очно-заочная (Объем пренебрежимо мал и не растет с увеличением БД)
Сумма	-	-	67172 E	-

Общий объём базы данных: $67172 \text{ E} = 67172 * 240 \sim 15,7 \text{ Mb} \text{ ###}$

Избыточность модели Фактический объём модели: 67172 E

Избыточными данными примем дублирующуюся информацию: таблицы для связи отношений “многие ко многим” и внешние ключи (FK): * **Faculty:** 0b * **Department:** faculty_name = 100b * **Educational Program:** direction_code + department_name = 200b * **Practice:** educational_program_name = 100b * **Training Plan:** educational_program_name = 100b * **Discipline:** 0b * **Discipline Curriculum:** discipline_name + department_name + annotation_name = 300b * **Discipline Annotation:** 0b * **Discipline of Training Plan:** training_plan_name + discipline_name = 104b * **Curriculum of Educational Program:** educational_program_name + discipline_curriculum_id = 104b * **Form of Study:** 0b * **Form of Program Study:** educational_program_name + form_of_study_name = 200b

Объем избыточных данных равен: 25 E + 200 E + 300 E + 300 E + 9000 E + 15600 E + 3120 E + 200 E = 28745 E

Текущий объём избыточных данных: ~6,7 Mb

“Чистый” объем данных: 67172 E - 28745 E = 38427 E

Отношение между фактическим объемом модели и “чистым” объемом данных: $67172/38427 = 1,75$ ### Направление роста модели

Как видно из таблицы с оценкой удельного объёма информации, стремительное увеличение объёма происходит тогда, когда появляются связи в таблицах, описывающих связи нереляционной модели - это таблицы Discipline of Training Plan и Curriculum of Educational Program.

Сценарий использования

- Получить список образовательных программ
SELECT educational_program_name FROM EducationalProgram LIMIT 25;
- Получить список учебных планов
SELECT training_plan_name FROM TrainingPlan LIMIT 25;
- Получить список рабочих программ дисциплин
SELECT discipline_name FROM Discipline LIMIT 25;

- Получить список дисциплин, которые есть и первом учебном плане, и во втором

```
SELECT discipline_name FROM DisciplineOfTrainingPlan
WHERE training_plan_name = "Учебный план № 338-22 (прием
2022)" AND discipline_name IN
(SELECT discipline_name FROM DisciplineOfTrainingPlan
WHERE training_plan_name = "Учебный план № 338-23 (прием
2023)")
LIMIT 25;
```

- Получить список дисциплин, которые присутствуют в одном учебном плане и отсутствуют в другом учебном плане

```
SELECT discipline_name FROM DisciplineOfTrainingPlan
WHERE training_plan_name = "Учебный план № 338-22 (прием
2022)" AND discipline_name NOT IN
(SELECT discipline_name
FROM DisciplineOfTrainingPlan
WHERE training_plan_name = "Учебный план № 338-23 (прием
2023)")
LIMIT 25;
```

- Получить список дисциплин на конкретной кафедре

```
SELECT discipline_name FROM DisciplineCirriculum
WHERE department_name = "МО ЭВМ"
LIMIT 25;
```

Сравнение моделей

Объем данных в Neo4j равен 55162 Е байт (~12,9 Mb); В SQL - 67172 Е байт (~15,7 Mb).

Избыточность модели Neo4j = 1. Избыточность модели SQL = 1.75.

Т. е., графовая модель БД занимает меньше памяти, чем реляционная БД.

Вывод

Таким образом, NoSQL модель лучше подходит для реализации данного проекта, так как он занимает меньше памяти и обладает меньшей избыточностью.

5. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

Краткое описание

Приложение позволяет просматривать образовательные программы, дисциплины, учебные планы различных направлений, а также имеется возможность сравнения по различным параметрам учебные планы и смотреть на результаты в таблице.

Front-end представляет собой SPA-приложение, с помощью которого можно авторизовываться, просматривать и фильтровать списки образовательных программ и учебных планов, а также есть возможность добавить новый учебный план и новую образовательную программу. Возможно посмотреть весь граф с учебными планами в пункте “Визуализация графа”.

Backend представляет собой сервер, на котором описаны различные пути для получения и обработки данных. Здесь происходит соединение с базой данных Neo4j.

Схема экранов приложения

Каталог учебных планов и программ ЛЭТИ

Создать аккаунт

Логин

Vladislav@gmail.com

Пароль

Войти

Регистрация

Рисунок 8 – Авторизация в приложении

Каталог учебных планов и программ ЛЭТИ

Экспорт БД

Выйти

Образовательные программы

Учебные планы

Сравнение учебных планов

Визуализация графа

Список образовательных программ

редактировать

добавить

начните печатать...

Математическое обеспечение программно-информационных систем

Системы искусственного интеллекта

Акустические приборы и системы

Биотехнические и медицинские аппараты и системы

Рисунок 9 – Просмотр списка образовательных программ

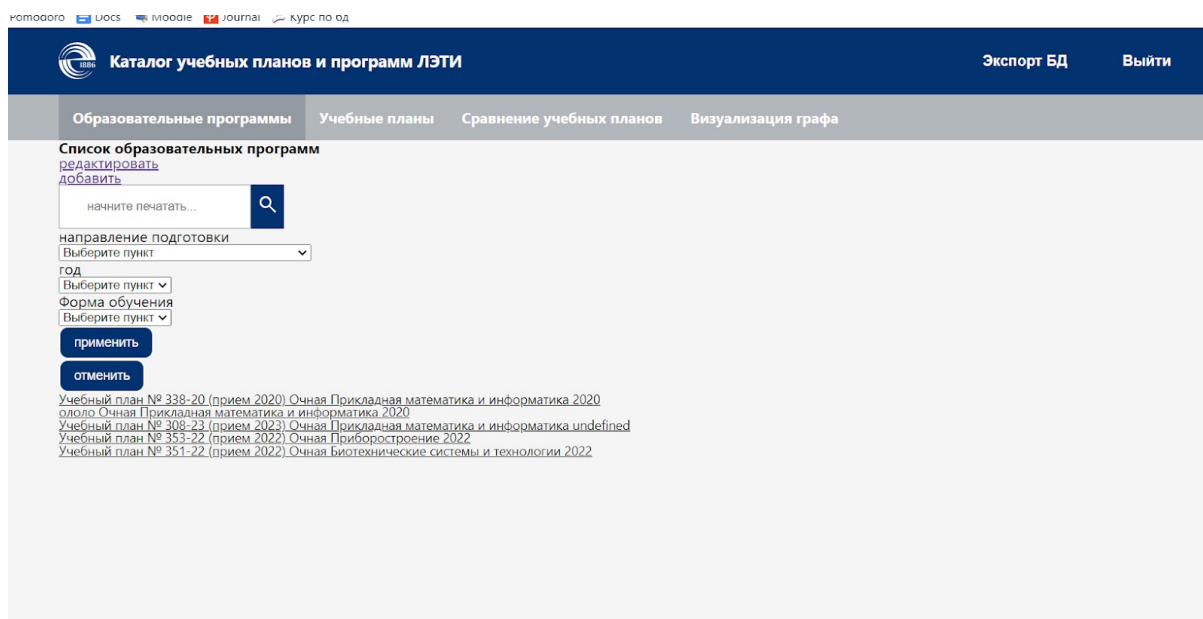


Рисунок 10 – просмотр и фильтрация списка учебных планов

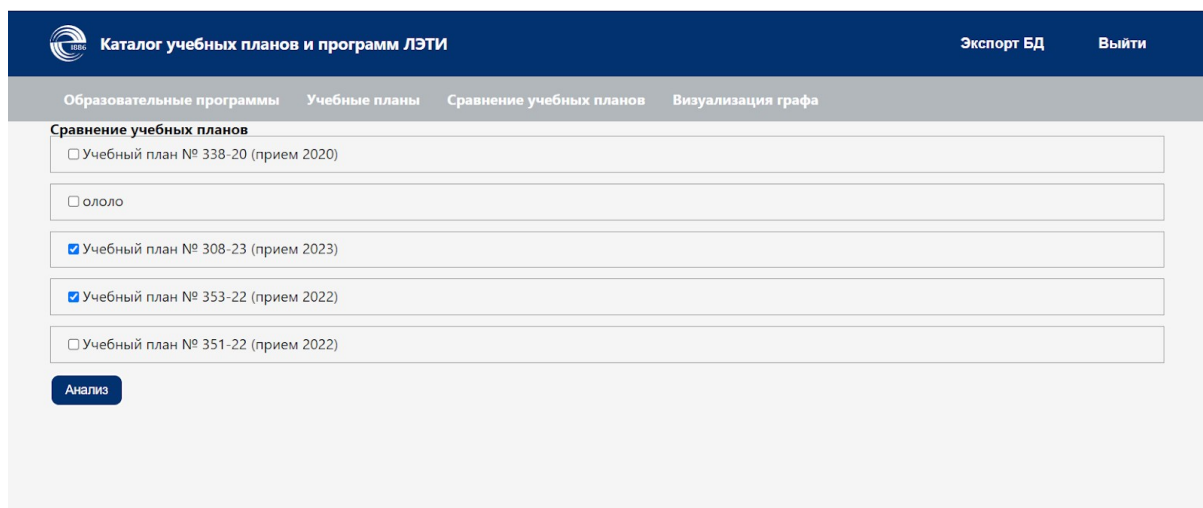


Рисунок 11 – Страница выбора учебных планов для сравнения



Рисунок 15 – Страница визуализации графа

Использованные технологии

Front-end: React, Typescript, HTML, Sass, JS/

Back-end: Node JS, Express

База данных: Neo4j

Ссылки на Приложение

1. Ссылка на GitHub: <https://github.com/moevm/nosql2h23-study-progs>

5. ВЫВОДЫ

Результаты

В ходе выполнения работы было разработано приложение для хранения, просмотра и редактирования учебных планов и программ университета “ЛЭТИ”. Приложение представляет возможность просматривать планы как в виде таблицы, так и в виде графа. С помощью приложения сравнивать учебные планы и добавлять информацию о новых.

Недостатки и пути для улучшения полученного решения

Недостатком является несколько ограниченный функционал: так, на данном этапе нельзя сравнивать более двух планов одновременно. Также для обеспечения работоспособности программы необходимо реализовать обработку ошибок на серверной части приложения.

Будущее развитие решения

В будущем планируется реализовать автоматическую загрузку информации об учебных планах и программах с сайта <https://etu.ru/sveden/education/eduop>. Также необходимо реализовать просмотр информации об учебных планах на страницах учебных планов.

6. ПРИЛОЖЕНИЯ

Инструкция по запуску программы:

- 1) Войти в проект
- 2) Открыть два терминала
- 3) В первом ввести команду `cd client`
- 4) Затем `npm start`
- 5) Во втором терминале ввести команду `cd server`
- 6) Запустить сервер `npm run start`

7. ИСПОЛЬЗУЕМАЯ ЛИТЕРАТУРА

1. Документация Neo4j. URL: <https://neo4j.com/docs/>

