МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

по дисциплине «Введение в нереляционные базы данных»

Тема: Анализ научных статей

Студенты гр. 5303 _____ Половинкин А.А.

гр. 5381 _____ Немтырева А.С.

_____ Филиппова В.А.

Преподаватель _____ Заславский М.М.

Санкт-Петербург

2018

ЗАДАНИЕ НА ПРОЕКТ

Студенты Немтырева А.С., Половинкин А.А., Филиппова В.А.

Группа 5381,5303

Тема проекта: Анализ научных статей

Исходные данные (технические требования):

Научная электронная библиотека КиберЛенинка

Содержание пояснительной записки:

«Содержание», «Введение», «Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки: не менее 10 страниц.

Дата выдачи задания: 14.09.2018

Дата сдачи проекта: 26.12.2018

Дата защиты проекта: 26.12.2018

Студенты гр. 5303	_____	Половинкин А.А.
гр. 5381	_____	Немтырева А.С.
	_____	Филиппова В.А.
Преподаватель	_____	Заславский М.М.

АННОТАЦИЯ

В данном проекте описываются этапы разработки web-приложения для анализа научных статей. Основное внимание приходится на взаимодействие приложения с нереляционной базой данных, в которой хранятся данные о научных статьях. В результате разработано действующее приложение.

SUMMARY

This project describes the stages of developing a web application for analyzing scientific articles. The focus is on the interaction of the application with a non-relational database, which stores data about scientific articles. As a result, a valid application has been developed.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель работы заключается в создании удобного веб-приложения для анализа научных статей.

Во время работы над научными статьями и проектами возникает необходимость хранить используемые публикации. Стандартный поход к этой задаче - хранить данные в древовидной структуре или списке. Такими структурами могут быть файловая система, файл с ссылками на статьи, закладки в браузере и т.д. Такой подход ограничен и приводит к путанице в документах и чрезвычайно сложному анализу предметной области исследования.

Альтернативой дереву и списку является более общая структура - ориентированный граф. Граф - это совокупность набора вершин и набора ребер между ними. В самом деле, каждая серьезная научная статья, прошедшая рецензирование и публикацию, содержит в себе ссылки (references) на используемы работы. Эти ссылки вмести со статьями можно рассматривать как граф, где каждая вершина - это статья, а ссылка из одной статьи на другую - это ребро между ними.

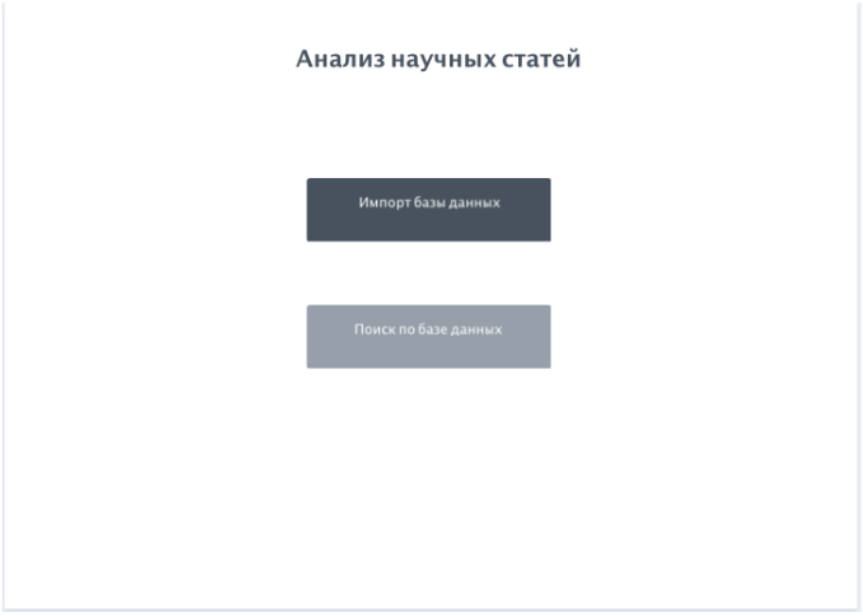
Также дополнительно каждая статья может относиться к различным категориям. Ими могут быть научные области, с которыми связана работа (биология, физика, компьютерная лингвистика и т.д.) или дата публикации.

1. Сценарии использования

1.1. Макет UI

UC -1 «Главная страница»

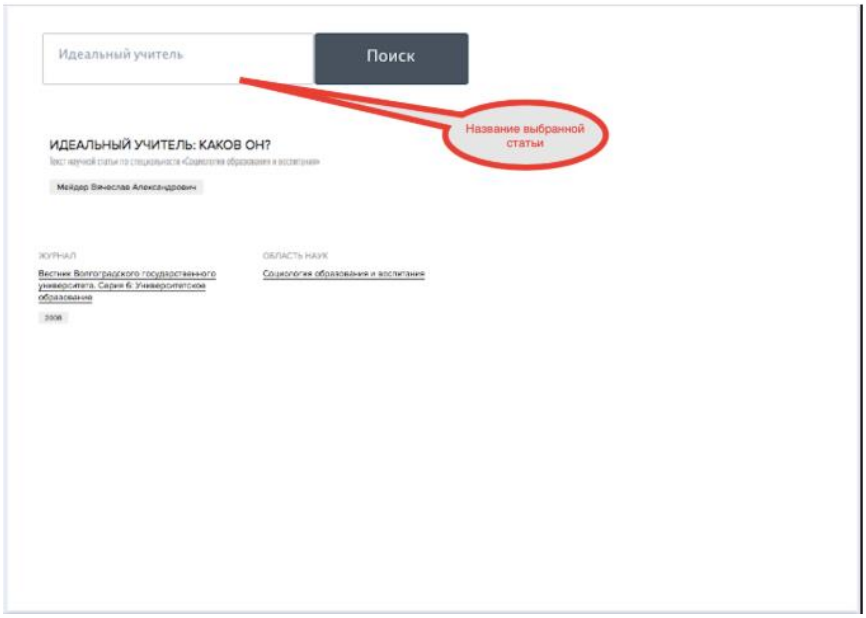
1. На экране перед пользователем находятся две кнопки : «Импорт базы данных», «Поиск по базе данных» Изначально активная только кнопка «Импорт базы данных»,после загрузки БД станет доступна кнопка «Поиск по базе данных»
2. При нажатии на кнопку «Импорт базы данных» произойдет добавление данных из внешних источников.
3. При нажатии на кнопку «Поиск по базе данных» откроется страница с поиском статьи по базе данных.



Экран 1

UC -2 «Поиск статьи»

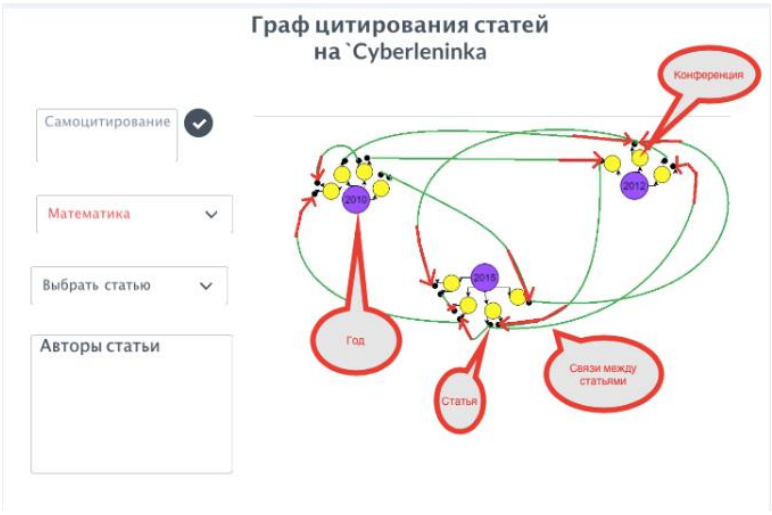
1. При нажатии на поле с поиском, пользователь вводит название статьи, выбирает нужную и нажимает кнопку поиск.
2. При нажатии на кнопку «поиск» на странице отображается краткая информация о статье.



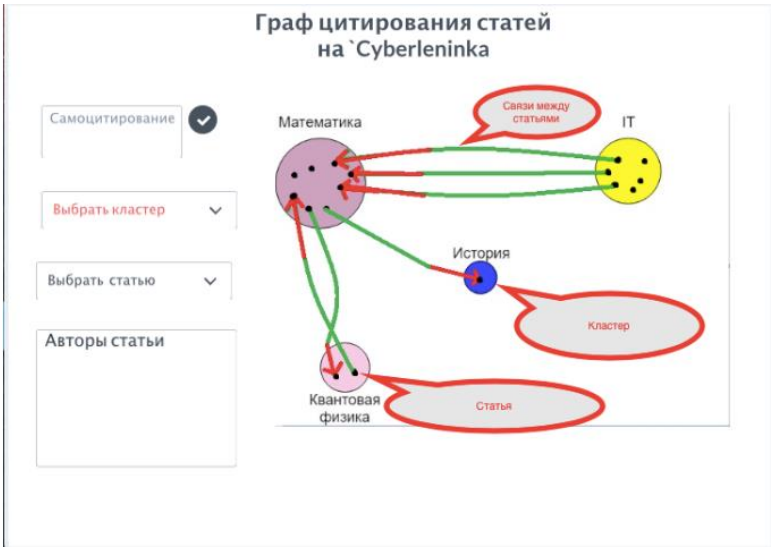
Экран 2

UC -3 «Граф цитирования статей»

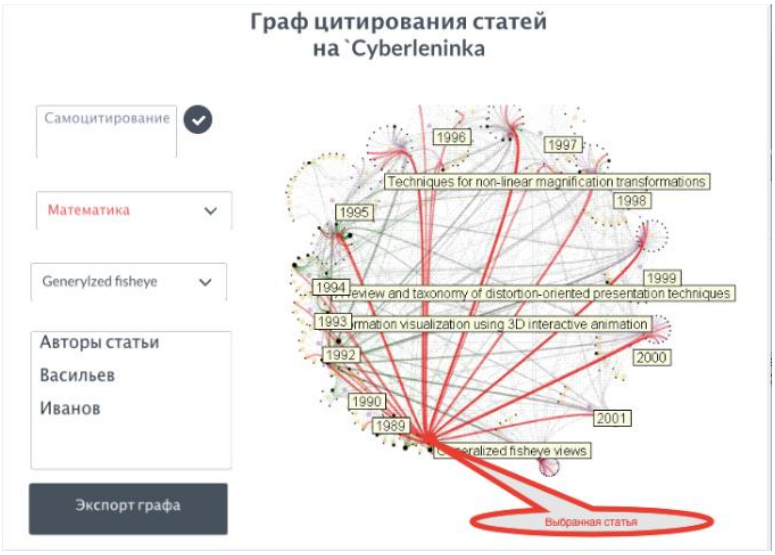
1. Пользователь в check-box включает/отключает "Самоцитирование"
2. Перед пользователем отображается граф в котором вершины это тематика статей,а ребра это ссылки на выбранную тематику со стороны других тематик
3. Пользователь в слайдере «Выбрать кластер» указывает нужную тематику статей – отображается граф цитирования для выбранной тематики; 3.1.Отображается граф в котором вершины фиолетового цвета отображают год,желтого-конференцию,черного- название публикации. Ребра-ссылки на выбранную статью со стороны других статей
4. Пользователь в слайдере «Выбрать статью» выбирает нужную для поиска цитирования статью; 4.1.Когда выявлены наиболее цитируемые публикации, пользователь может выбрать любую такую публикацию и просмотреть дополнительную информацию о ней: название, авторов, какие другие публикации на нее ссылаются. Отображается граф для выбранной пользователем публикации Ребра-ссылки на выбранную публикацию со стороны других публикаций показаны красным.
5. Пользователь в поле «Авторы статьи» отображаются авторы статей
6. После указания тематики и названия статьи,пользователю будет доступна кнопка "Экспорт графа" для вывода графа из текущей базы данных во внешний источник



Экран 3



Экран 4



Экран 5

1.2. Сценарии использования готового приложения

1.Главная страница приложения

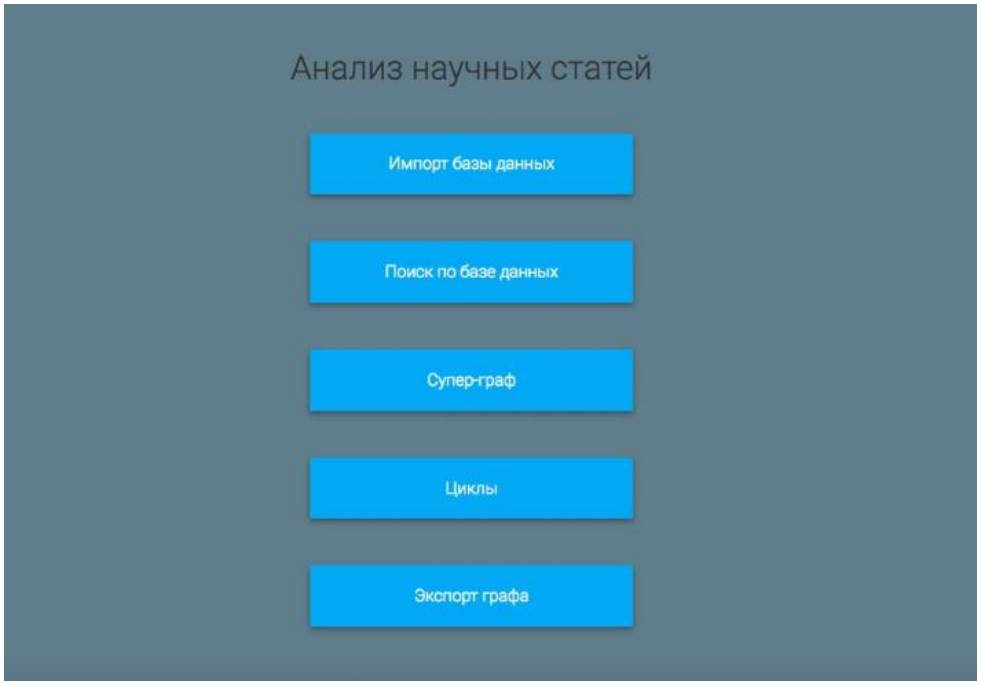


Рис.1

2.Действие	Результат
Пользователь на главной странице нажимает на кнопку «Импорт базы данных»	Перед пользователем появляется пустое окно для ввода данных о научной статье

Рис.2

3.Действие	Результат
Пользователь вводит данные	Появляется сообщение об успешном импорте

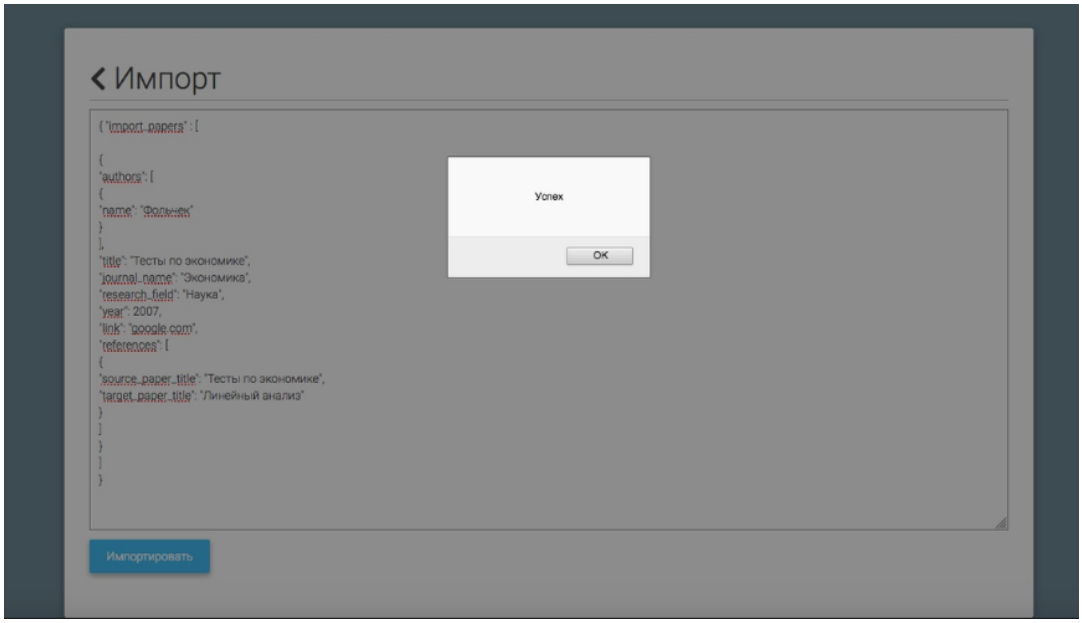


Рис.3

4.Действие	Результат
Пользователь на главной странице нажимает кнопку «Поиск по базе данных»	Открывается соответствующая страница
Пользователь вводит название статьи в поисковую строку	На экран выводится информация о научной статье (тема статьи, журнал и год публикации)

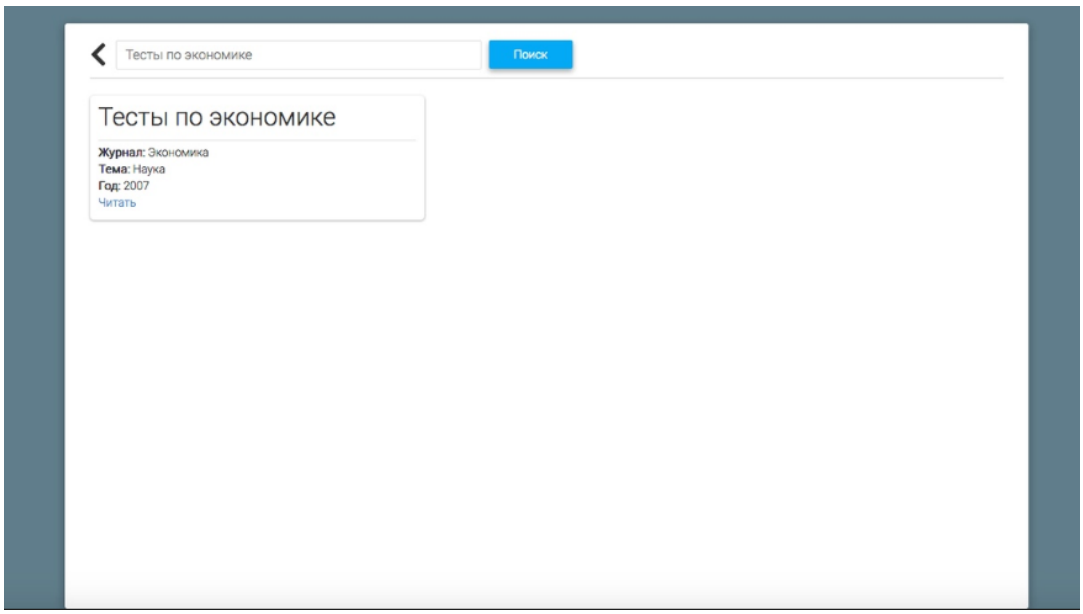


Рис.4

5.Действие	Результат
Пользователь нажимает на статью	Выводится граф цитирования для данной статьи

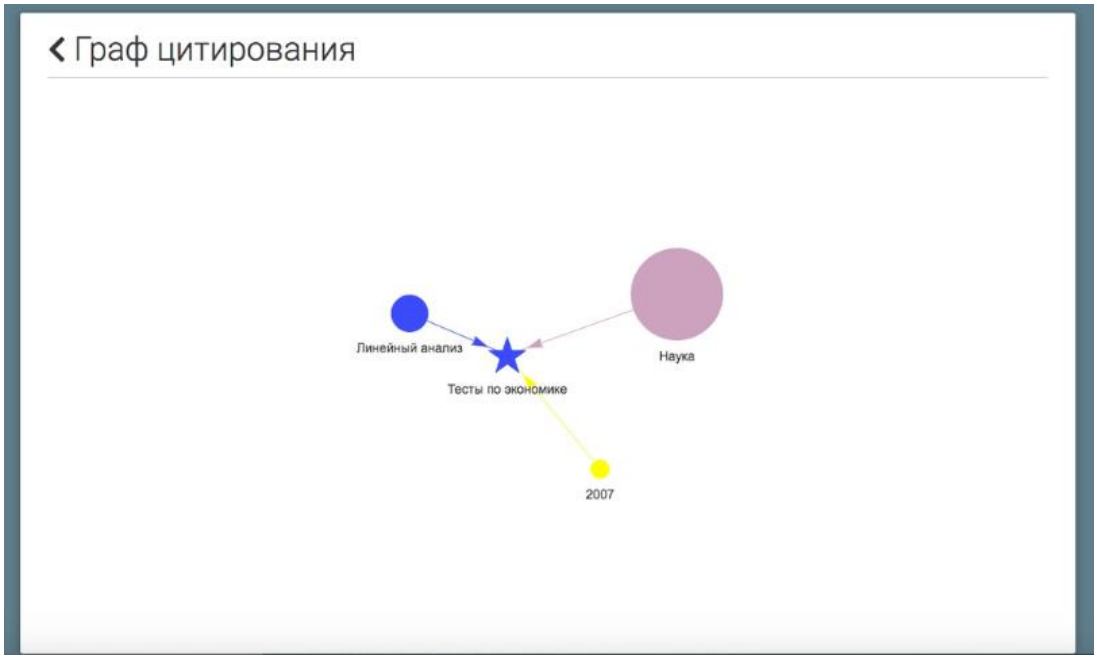


Рис.5

6.Действие	Результат
Пользователь на главной странице нажимает кнопку «Супер-граф»	Открывается страница с графом цитирования, где отображены связи статей с журналами, годами публикаций и тематикой.

Легенда :

- Синий круг - статья
- Желтый круг – журнал
- Оранжевый круг – тема
- Фиолетовый круг - год

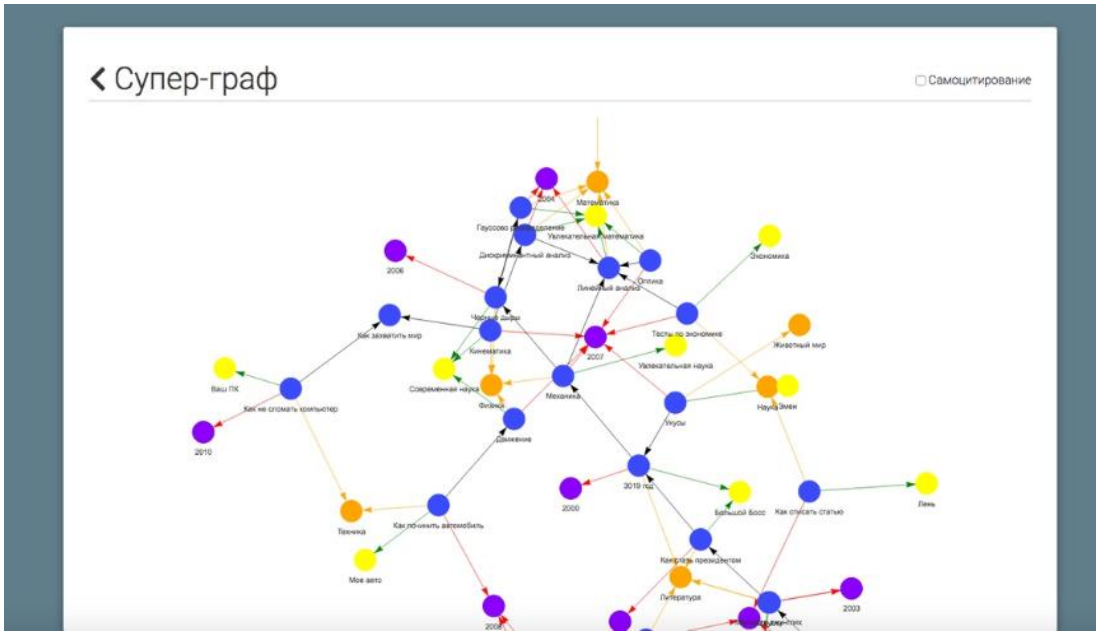


Рис.6

7.Действие	Результат
Пользователь выбирает на графе определенную тему (оранжевый круг) (Н-р: Литература)	Открывается страница с графом цитирования статей связанных данной тематикой

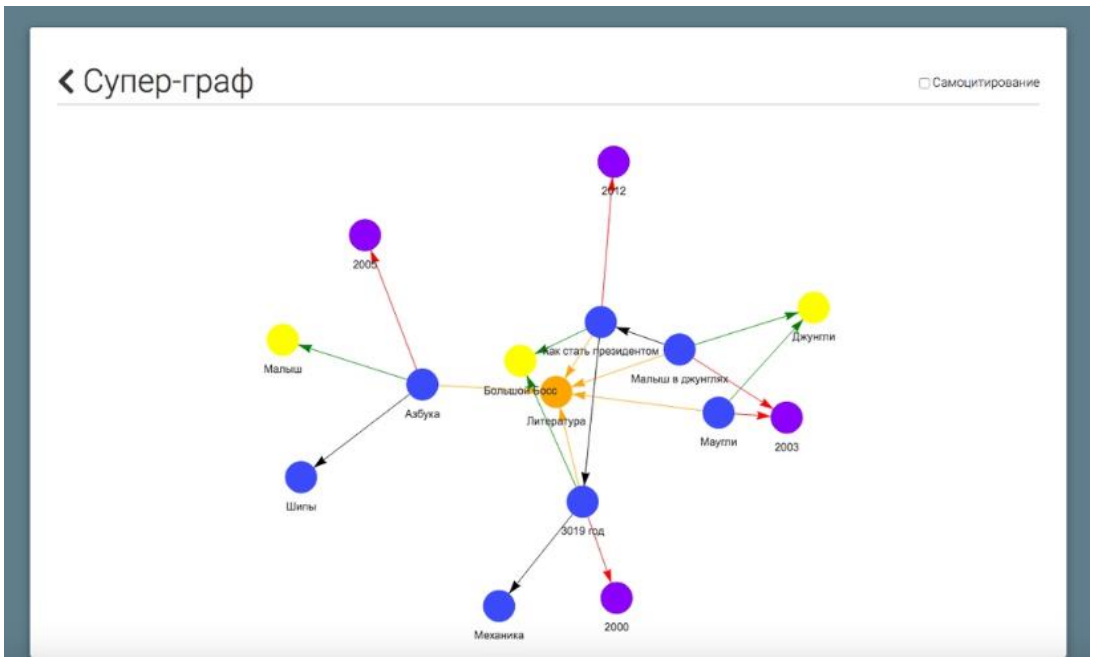


Рис.7

8.Действие	Результат
Пользователь выбирает на графе определенную статью (Н-р: Азбука)	Открывается страница с графом цитирования для данной статьи.

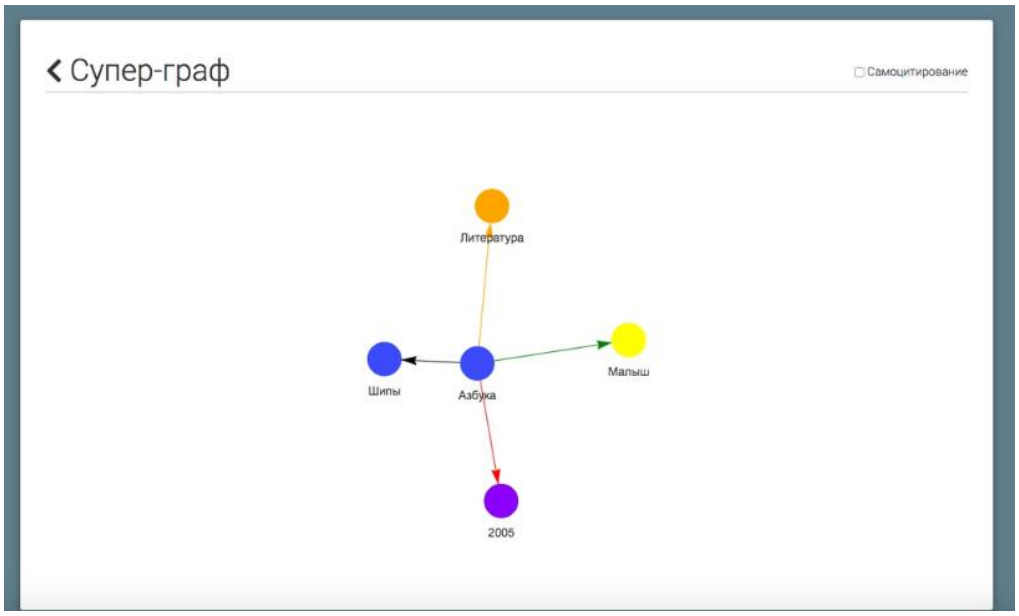


Рис.8

9.Действие	Результат
Пользователь на экране «Супер-граф» выключает «Самоцитирование»	Открывается страница с графом цитирования, где отображены связи статей с журналами, годами публикаций и тематикой, но не отображаются связи, где у статей ссылающихся друг на друга совпадает автор

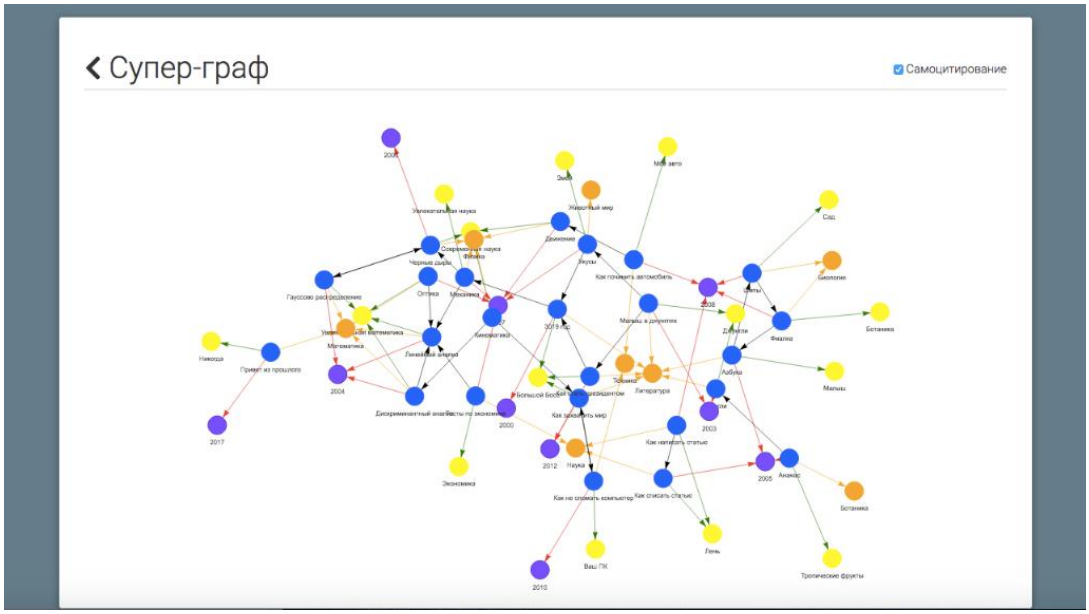


Рис.9

10.Действие	Результат
Пользователь на главной странице нажимает кнопку «Циклы»	Открывается страница с циклами цитирования

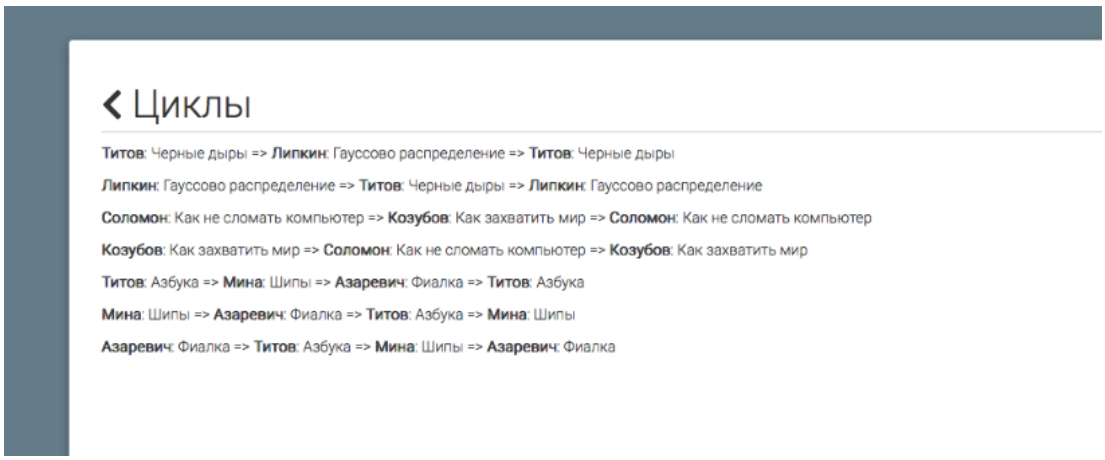


Рис.10

11. Действие	Результат
Пользователь на главной странице нажимает кнопку «Экспорт графа»	Открывается страница для сохранения БД в выбранном формате

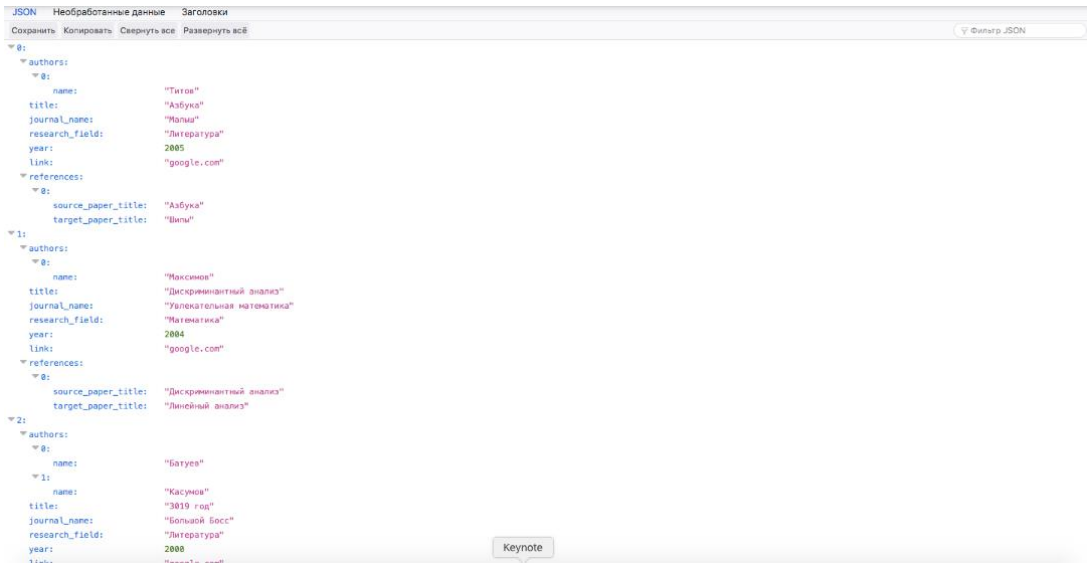


Рис.11

12.Действие	Результат
Пользователь на экране «Экспорт графа» нажимает кнопку сохранить	Сохранение БД в выбранном формате

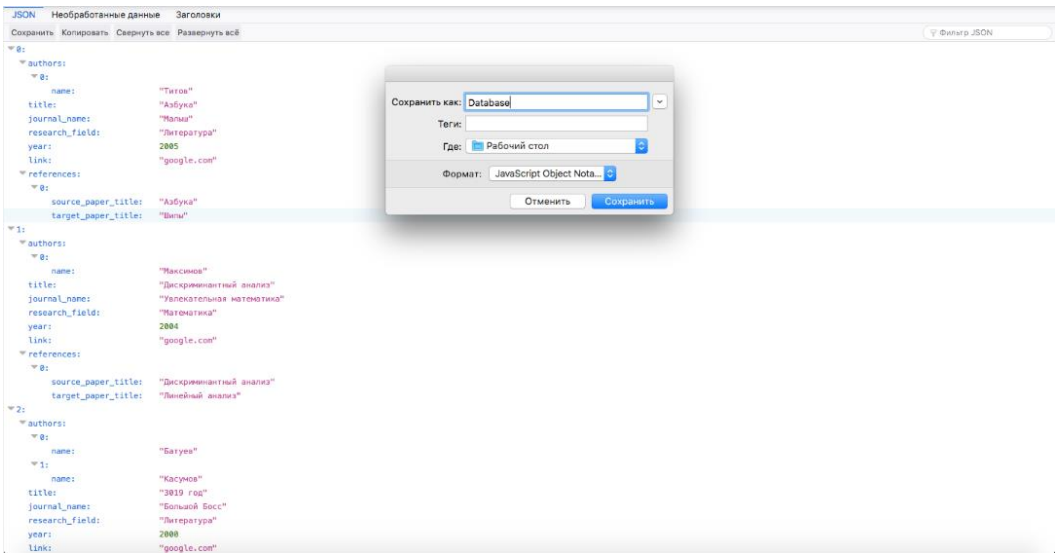


Рис.12

2. Модель данных

2.1. Описание структуры

На сайте cyberleninka.ru есть научные статьи. Из страницы с научной статьей можно выделить следующие данные:

- Название/заголовок научной статьи
- Авторы статьи
- Год издания статьи в журнале
- Кол-во просмотров страницы со статьей
- Кол-во загрузок статьи
- Журнал
- Область наук
- Теги
- Похожие темы
- Текст научной статьи
- Список литературы

Было решено выделить следующие сущности:

- Автор статьи

```
name = ~32 байт
```

- Научная статья

```
title = ~100 байт
journal_name = ~ 38 байт
research_field = ~ 38 байт
year = 2 байта
link = ~ 126 байт
```

Автор статьи = ~ 32 байта

Научная статья = ~ 304 байта

Со связями:

- из автора в статью (статья x была написана автором y) = 0 байт
- из статьи в статью (в статье x цитируется статья y) = 0 байт

Так как связи графа neo4j мы оценили в 0 байт, а в sql связи хранятся в виде кросс-таблицы и одна запись занимает место равное сумме занимаемого места первичных ключей в двух таблицах умноженное на количество ребер.

```
(PK1 + PK2) * EDGE
```

То в таком случае neo4j выигрывает SQL.

2.2. Оценка объема в зависимости от количества статей

Для хранения N статей каждая из которых имеет по пять связей-ссылок на другие статьи и по 2 автора потребуется:

*N * paper_size + 5 * N * reference_edge_size + 2 * N * author_edge_size + (N / 2) * author_size*

Для Neo4j reference_edge_size и author_edge_size приняли равным нулю размер *бд = N * paper_size + (N / 2) * author_size*

Для SQL *N * paper_size + 5 * N * reference_edge_size + 2 * N * author_edge_size + (N / 2) * author_size*

SQL будет затрачивать больше памяти.

2.3. Запросы к модели

- Получение научной статьи по названию

```
MATCH (p:Paper { title: $title })
RETURN p
```

- Получение циклических цитирований по заданным фильтрам

```
MATCH p = (n:Paper)-[*]->(n:Paper) RETURN nodes(p)
```

- Добавление научной статьи

```
MERGE (
    p:Paper {
      title: $title,
      journal_name: $journal_name,
      research_field: $research_field
      year: $year
      link: $link
    }
  )
```

- Добавление автора

```
MERGE (a:Author { name: $name })
```

- Добавление связи WROTE

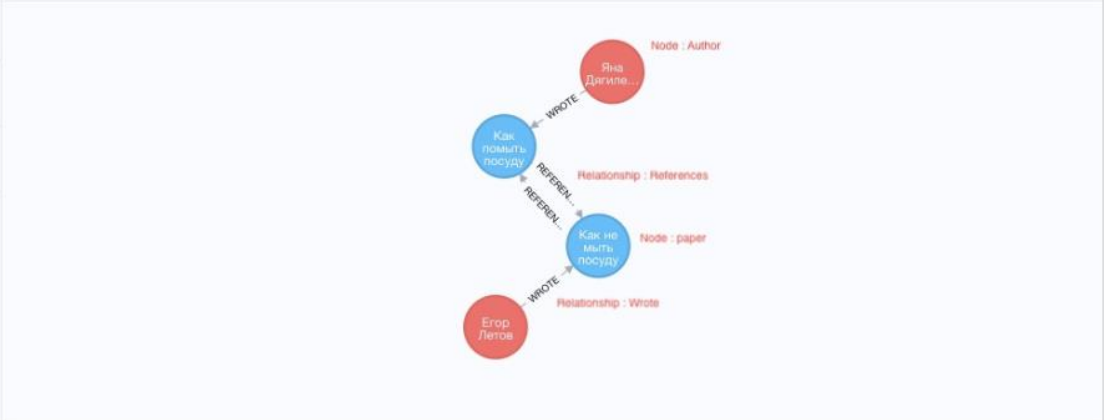
```
MATCH
(a:Author { name: $name }),
(p:Paper { title: $title })
MERGE (a)-[:WROTE]->(p)
```

- Добавление связи REFERENCES

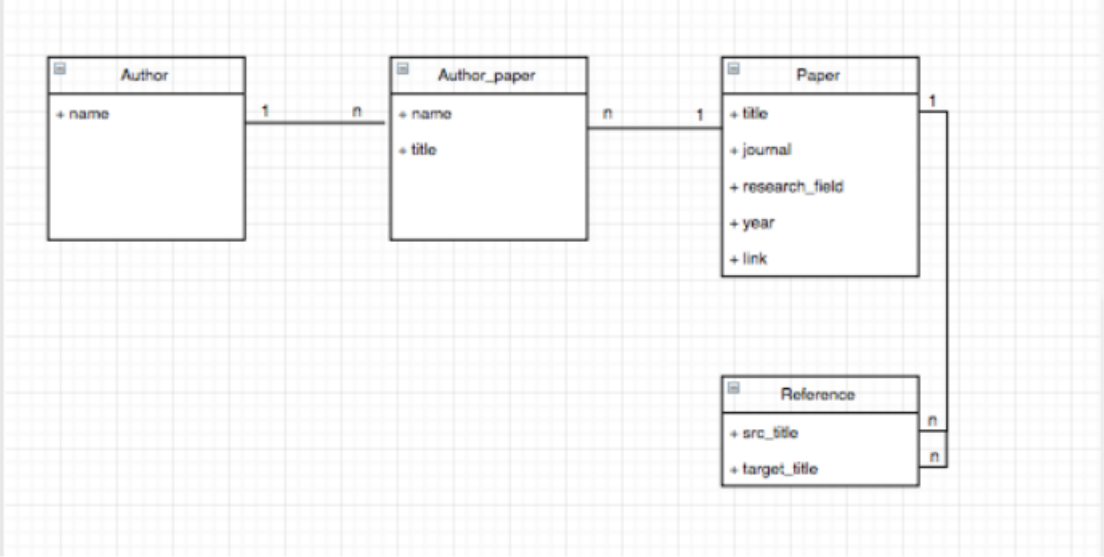
```
MATCH
(a:Paper { title: $title1 }),
(p:Paper { title: $title2 })
MERGE (p1)-[:REFERENCES]->(p2)
```

2.4. Графическое представление базы данных

- **Нереляционной БД**



- **Реляционной БД**



2.5. Сложность запросов

	Neo4j	SQL
Добавление автора	1 запрос	1 запрос
Добавление статьи	1 запрос	1 запрос
Добавление отношения цитирования	1 запрос	1 запрос
Добавление отношения написания	1 запрос	1 запрос
Получение автора	1 запрос	1 запрос
Получение статьи	1 запрос	1 запрос
Нахождение всех циклов	1 запрос	1 запрос с помощью например PSSQL иначе нельзя

Вывод:

Затраты памяти асимптотически равны Поиск циклов в графе будет вычисляться не оптимально, в несколько запросов, в neo4j запрос один Когнитивная сложность: очень просто допустить ошибку в запросе на поиск цикла в графе SQL. В neo4j ошибку допустить практически невозможно.

3. Разработанное приложение

Данное web-приложение направленно на анализ научных статей электронной библиотеки КиберЛенинка. В нём содержится 12 экранов. В приложении реализованы сервер и клиент. Осуществлено взаимодействие с базой данных Neo4j. Все запросы производятся со стороны сервера.

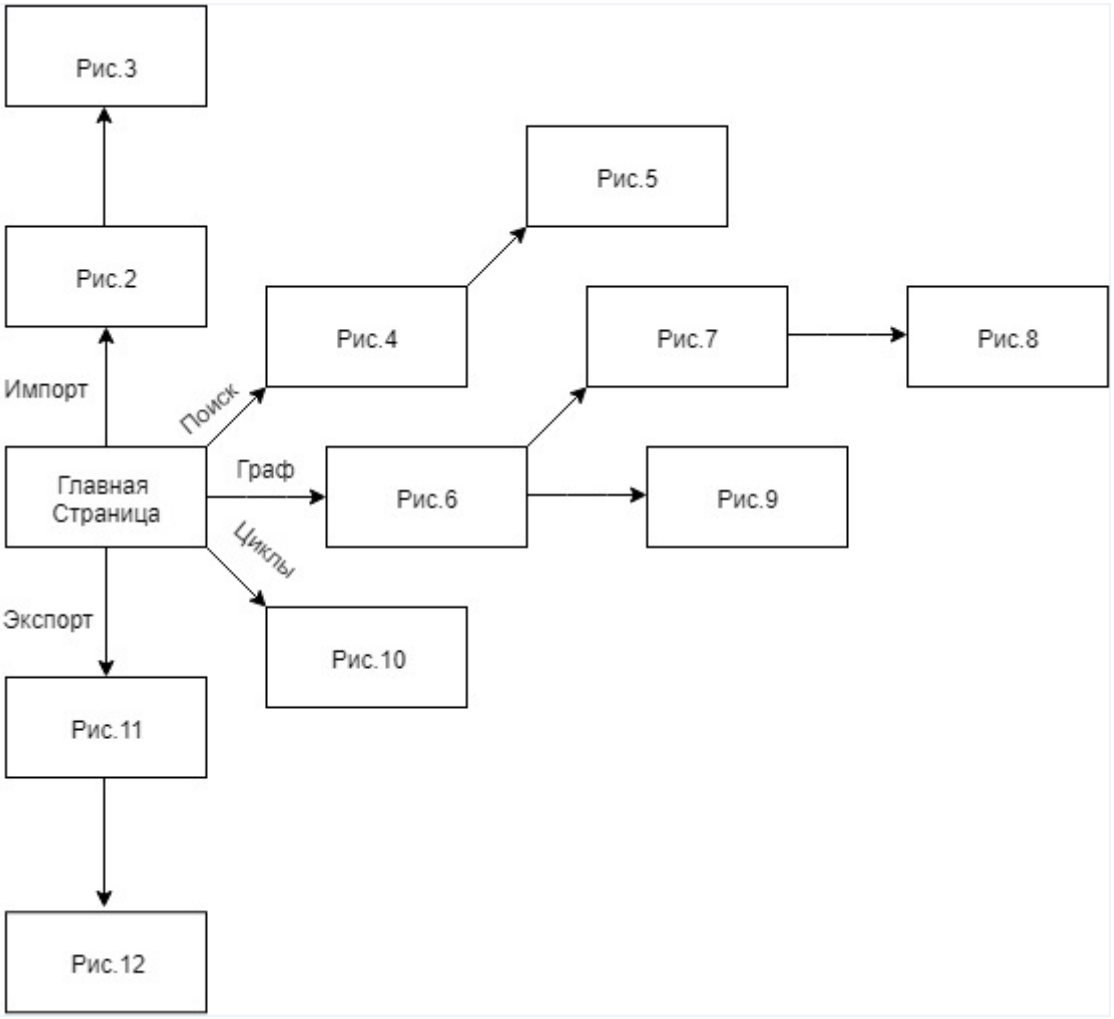
Использованные технологии:

1.Дизайн страниц – CSS

2.Динамический контент – JavaScript

3.СУБД – Neo4j

Схема экранов приложения:



Заключение

В результате выполнения работы было создано клиент-серверное приложение для анализа научных статей.

Недостатки полученного решения:

1) Строки склеиваются обычной интерполяцией, возможны NoSQL - инъекции .

2) Плохое регулярное выражение для парсинга элемента списка литературы, небольшое число срабатываний.

3) В Neo4j нельзя делать много запросов в рамках одного обращения к БД без костылей (WITH 1 as dummy), но возможно с помощью плагинов.

Будущее развитие решения:

1) Вместо драйвера использовать обычные http запросы в БД.

2) В Scale есть возможность написания своего интерполятора строк, в котором можно описать логику экранирования, внешне это будет выглядеть как обычная

интерполяция.

3) Улучшить регулярное выражение.

4) Переписать импорт с «WITH 1 as dummy» на вызов функции плагина «APOC»

Ссылка на приложение : https://github.com/moevm/nosql2018-paper_analysis

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Working with Data // Neo4j. URL : https://neo4j.com/developer/working-with-data/
2. Graph Visualization for Neo4j //Neo4j. URL : https://neo4j.com/developer/guide-data-visualization/
3. Docker // habr. URL : https://habr.com/company/southbridge/blog/428708/