Компонент веб-доступа

# Часть 1. Задача и основные возможности

Задача веб-компонента – предоставлять результаты работы системы пользователю в удобном для него виде.

Основные функции:

* поиск публикаций, авторов и областей знаний по части названия
* поиск наиболее цитируемых экспертов (авторитетов) в предметной области (отображается на панели управления);
* поиск новых зарождающихся областей исследований, так же как и тех, к которым интерес уже пропал (отображается на панели управления);
* просмотр различных индивидуальных параметров профиля автора публикаций и параметров его индивидуальной работы (количество публикаций, количество ссылок на них, дата последней публикации);
* просмотр различных индивидуальных параметров некоторой предметной области
* просмотр различных индивидуальных параметров конкретной публикации (предметные области, количество цитирований);
* произвольный запрос к графу знаний в терминах языка запросов графовой базы данных, используемой в СКА
* просмотр состояния баз данных проекта
* разделение доступа пользователей к информации

# Часть 2. Структура компонента. Основные технологии и инструменты

Компонент веб-доступа состоит из 2х частей:

- **sca\_web\_front** – часть, отвечающая за интерфейс, который пользователь видит в браузере. Она принимает ввод пользователя, отправляет на серверную часть Компонента веб-доступа, принимает ответ от него.

**sca\_web\_front** разрабатывается на Node.js 8.11.12 (<https://nodejs.org/en/>) с использованием библиотеки создания интерфейсов пользователя React

(<https://reactjs.org> )

Для работы используется менеджер пакетов Yarn 1.7.0 ( https://github.com/yarnpkg )

- **sca\_web\_back** – серверная часть, предоставляющая web API для работы интерфейсной части.

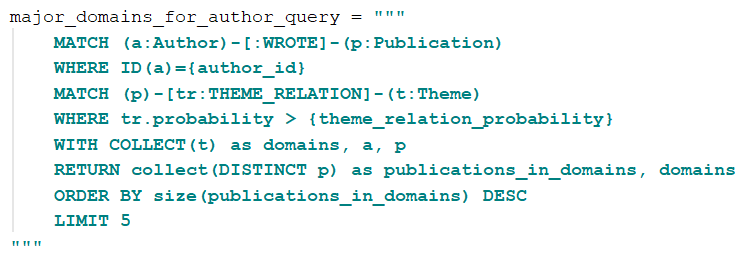
Она отвечает за предоставление данных только авторизованным пользователям и она, принимая сигнал от веб-интерфейса, делает запросы в БД Neo4j, подготавливает и отправляет результат на веб-интерфейс. Внешнему пользователю нет возможности обратиться к Neo4j, минуя требующий определенных ключей для входа сервер.

**sca\_web\_back** создается на ЯП Python 3.6 при помощи фреймоворков Django (https://www.djangoproject.com/ ) и Django REST framework (https://www.django-rest-framework.org/ ).

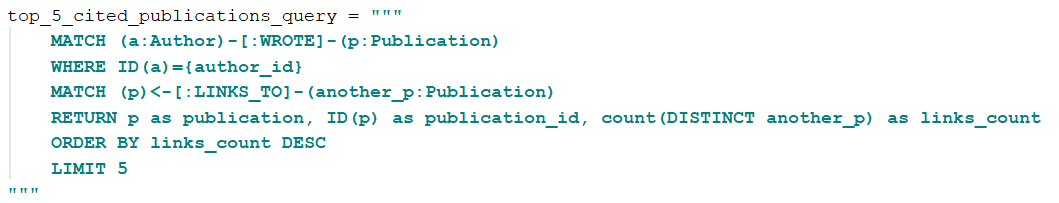
Для установления соединения и общения с БД Neo4j используется py2neo (<https://py2neo.org/v4/> ). Чтобы дополнительно обезопасить базу данных от несанкционированных действий веб компонент обращается к ней от имени специально созданного в БД пользователя с правами только на чтение.

# Часть 3. Работа серверной части, основные запросы к БД Neo4j

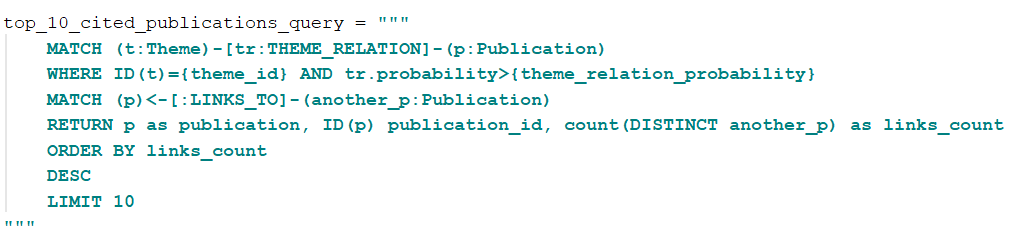
Основой взаимодействия серверной части компонента выдачи и БД Neo4j является класс NeoQuerier. Для обращения к БД он использует средства библиотеки *py2neo* (<https://py2neo.org/v4/>). Его задача – предоставить методы, которые будут возвращать требуемые результаты без необходимости устанавливать соединение и непосредственно писать запросы к БД на специальном языке. Ниже будут приведены некоторые его запросы и описание того, что они возвращают.



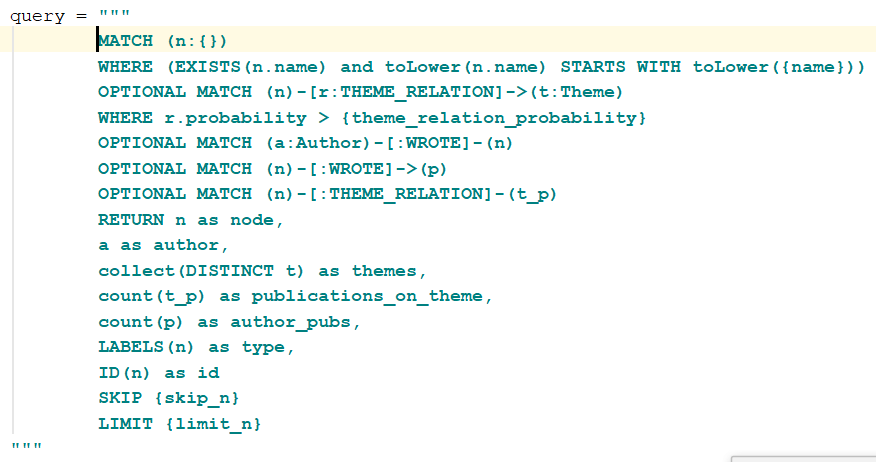
Листинг 4.1 – запрос на выборку 5 областей, в которых автор имеет наибольшее число публикаций



Листинг 4.2 – запрос на 5 наиболее цитируемых публикаций конкретного автора



Листинг 4.3 – запрос на 10 наиболее цитируемых публикаций конкретной области знаний



Листинг 4.4 – запрос общего поиска по части имени. В зависимости от типа найденного узла возвращает дополнительные данные для него

# Часть 4. Интерфейс пользователя и примеры работы с компонентом

Для работы с компонентов выдачи необходимо пройти авторизацию (рис 4.1)

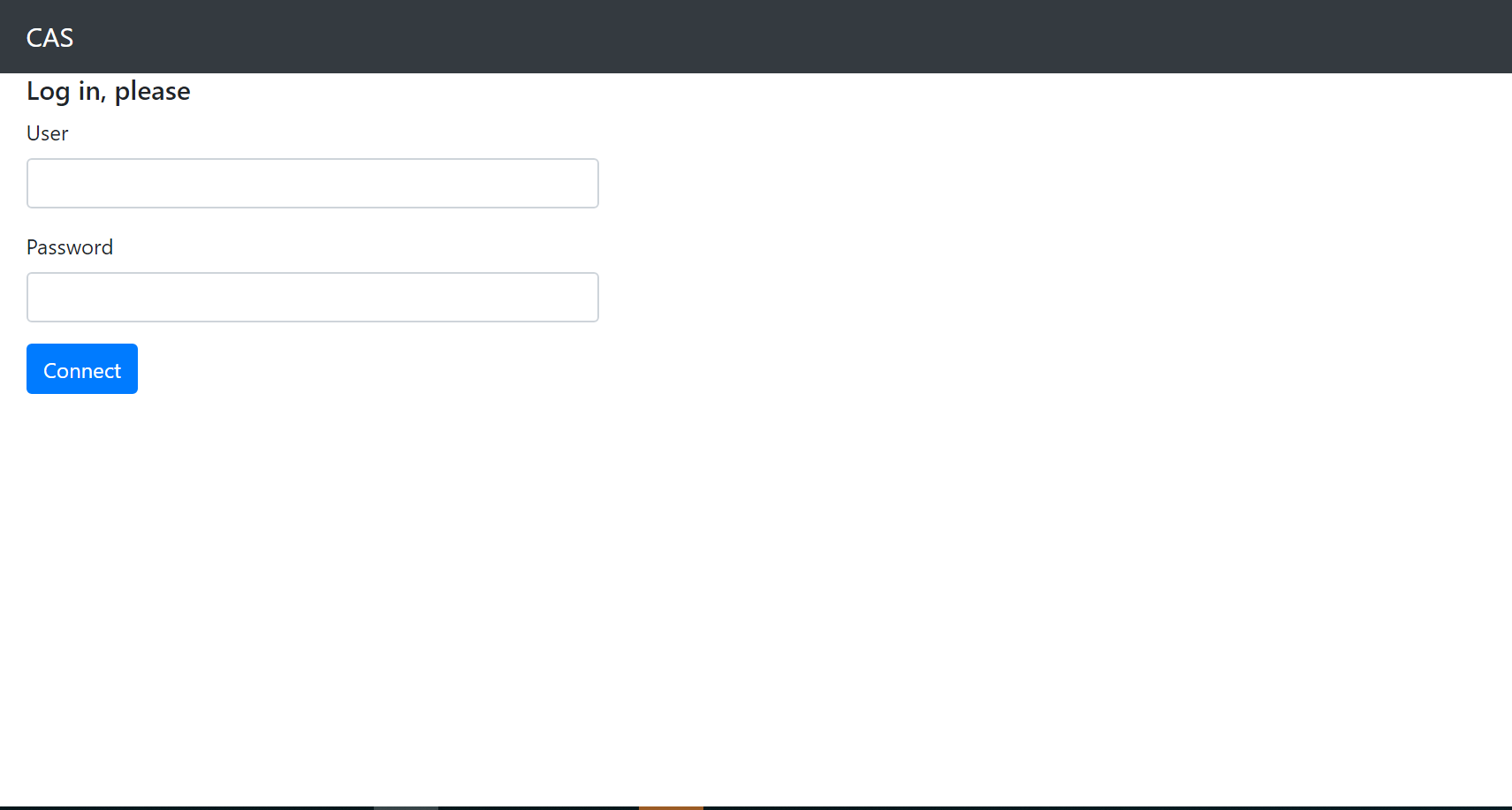


Рис 4.1 – приглашение к авторизации

Разделение доступа основано на использовании стандартной системы авторизации пользователей в Django Framework. Управление пользователями производится через стандартную панель администратора Django (рис 4.2). На данный момент используется 2 вида пользователей – привилегированные (Stuff) и непривилегированные.

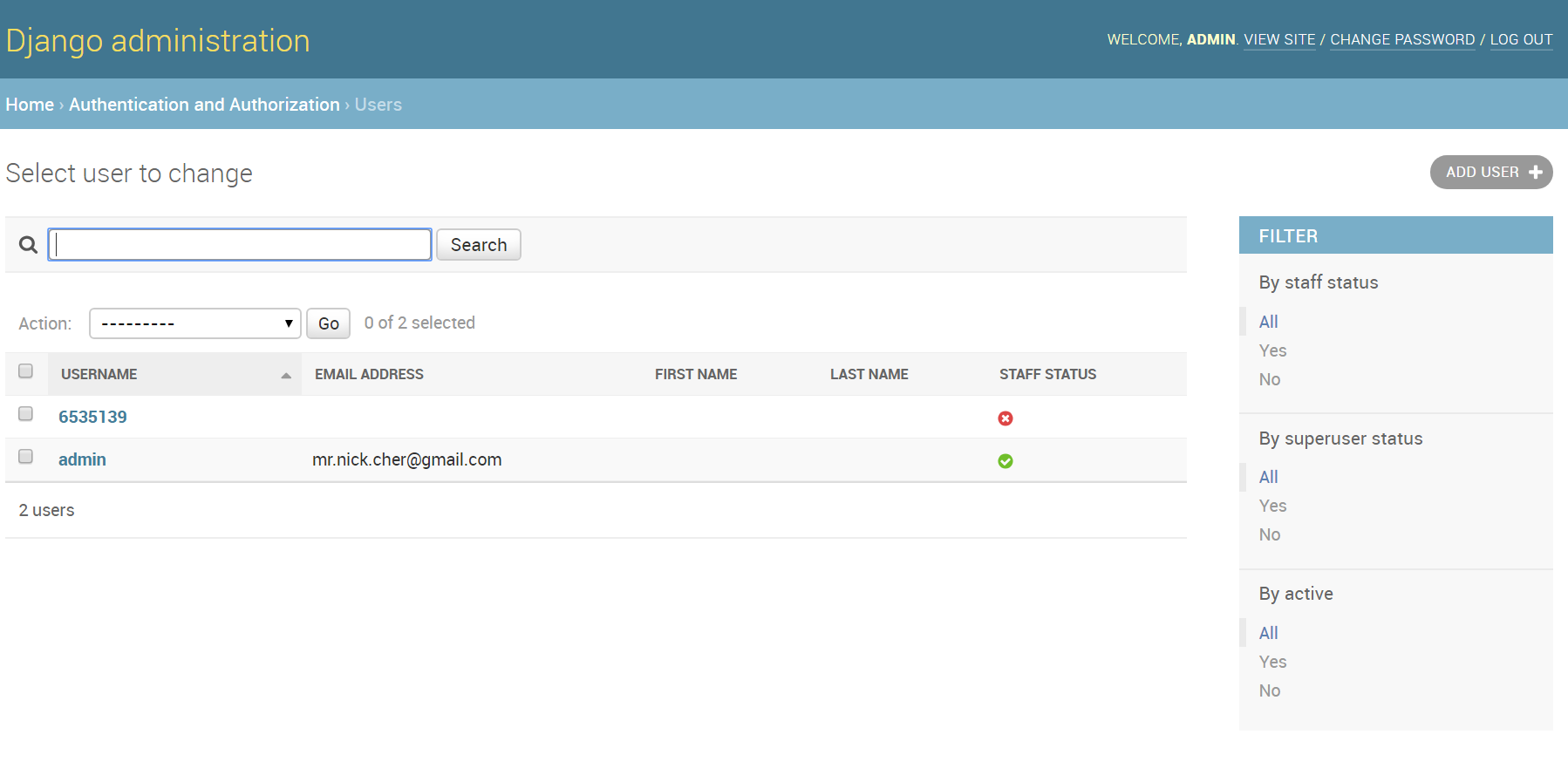


Рис 4.2 – панель администратора

Непривилегированным пользователям не доступны просмотр состояния баз данных и ввод запросов на языке графовой базы в специальном конструкторе.

На главной странице присутствую строка поиска публикаций, авторов и областей исследований по части названия и кнопки для перехода к более узким вариантам поиска:

поиск публикаций по ключевым словам, исследование областей знаний по их популярности (на основе количества публикаций в областях, уровня цитируемости публикаций), поиск экспертов (авторитетных авторов) в указанных областях и конструктор запросов, позволяющий работать с БД на языке Cypher. (См. рис. 4.3).

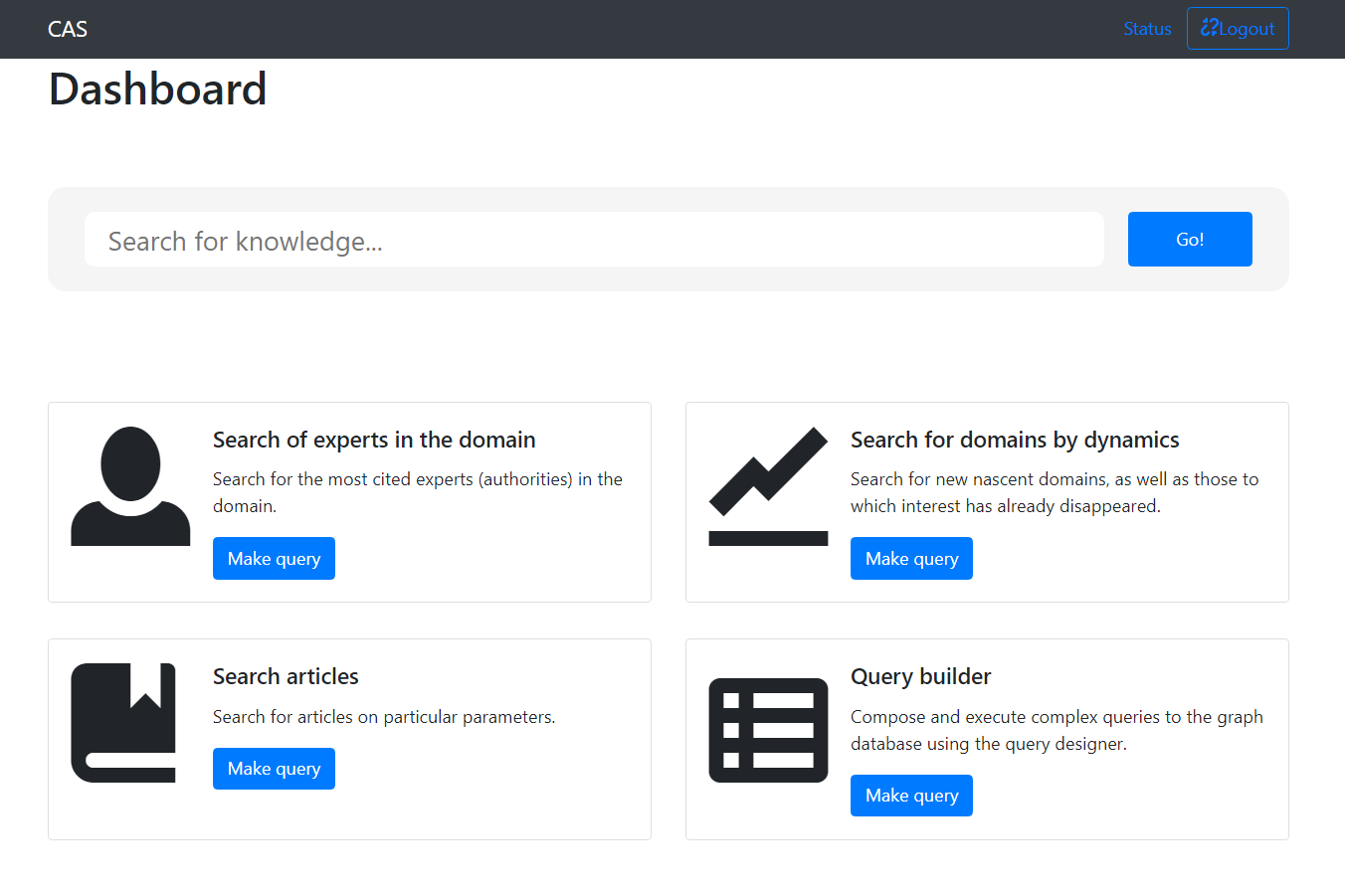


Рис 4.3 – главная страница выдачи

Поиск позволяет найти публикацию/ автора/ область знаний по части названия и затем просмотреть более подробную информацию о найденном.

Результаты поиска подгружаются по мере необходимости, по 10 за раз. Присутствует фильтр результатов – только публикации, только авторы, только области.

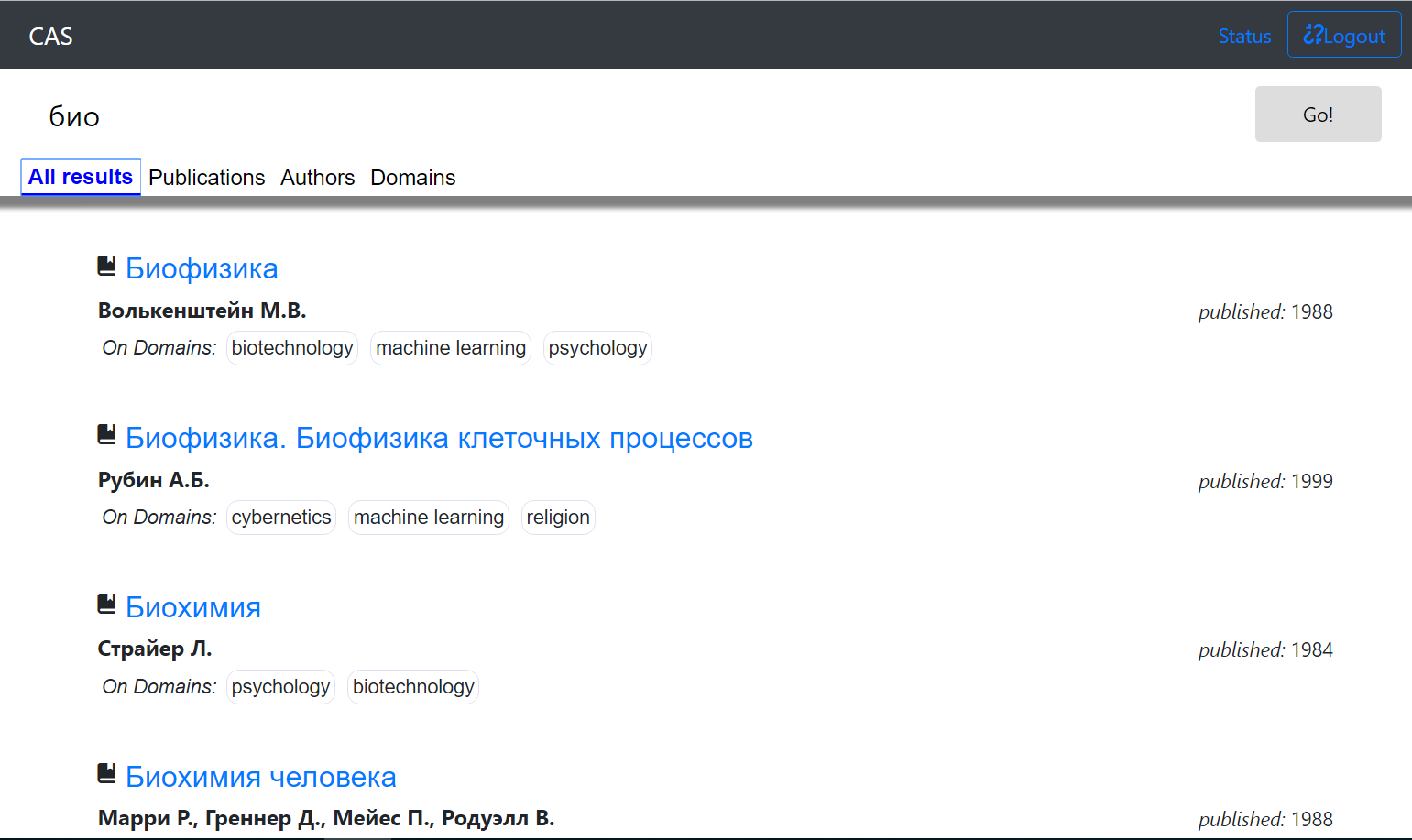


Рис 4.4. Все результаты

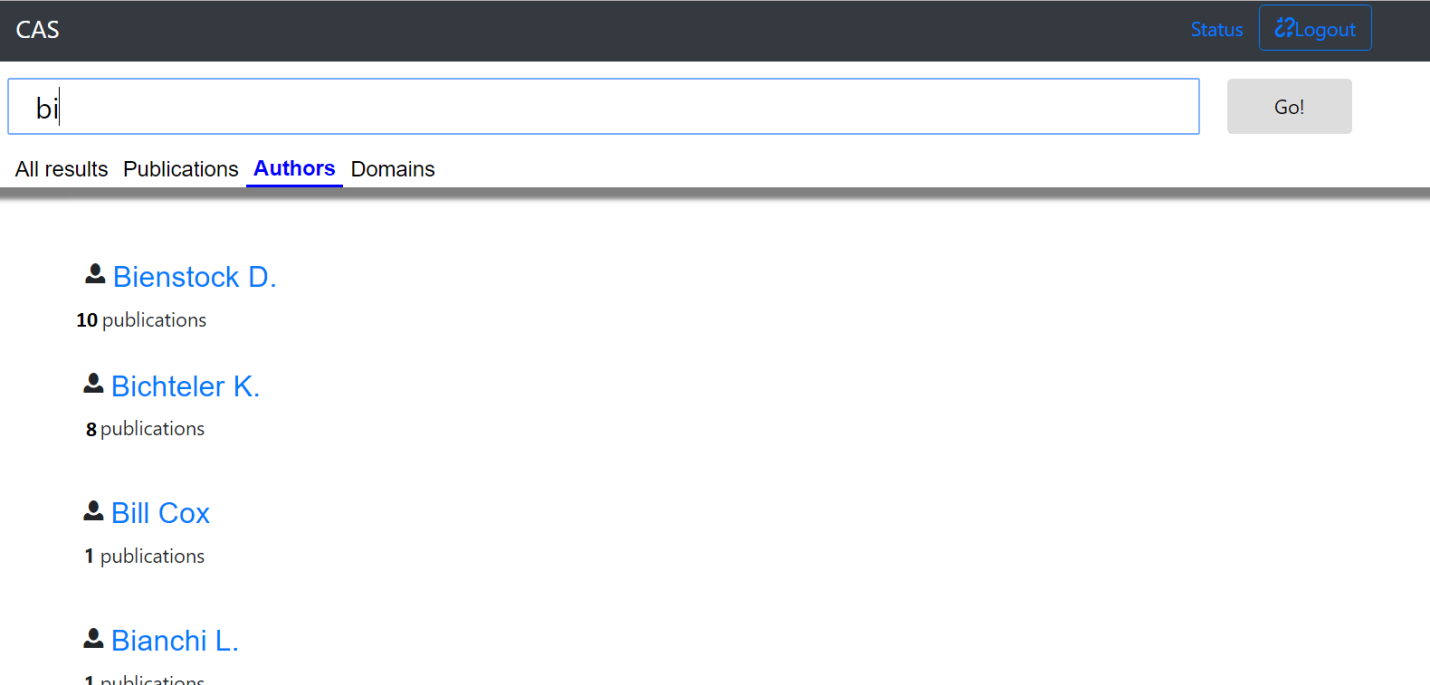


Рис 4.5 – только авторы

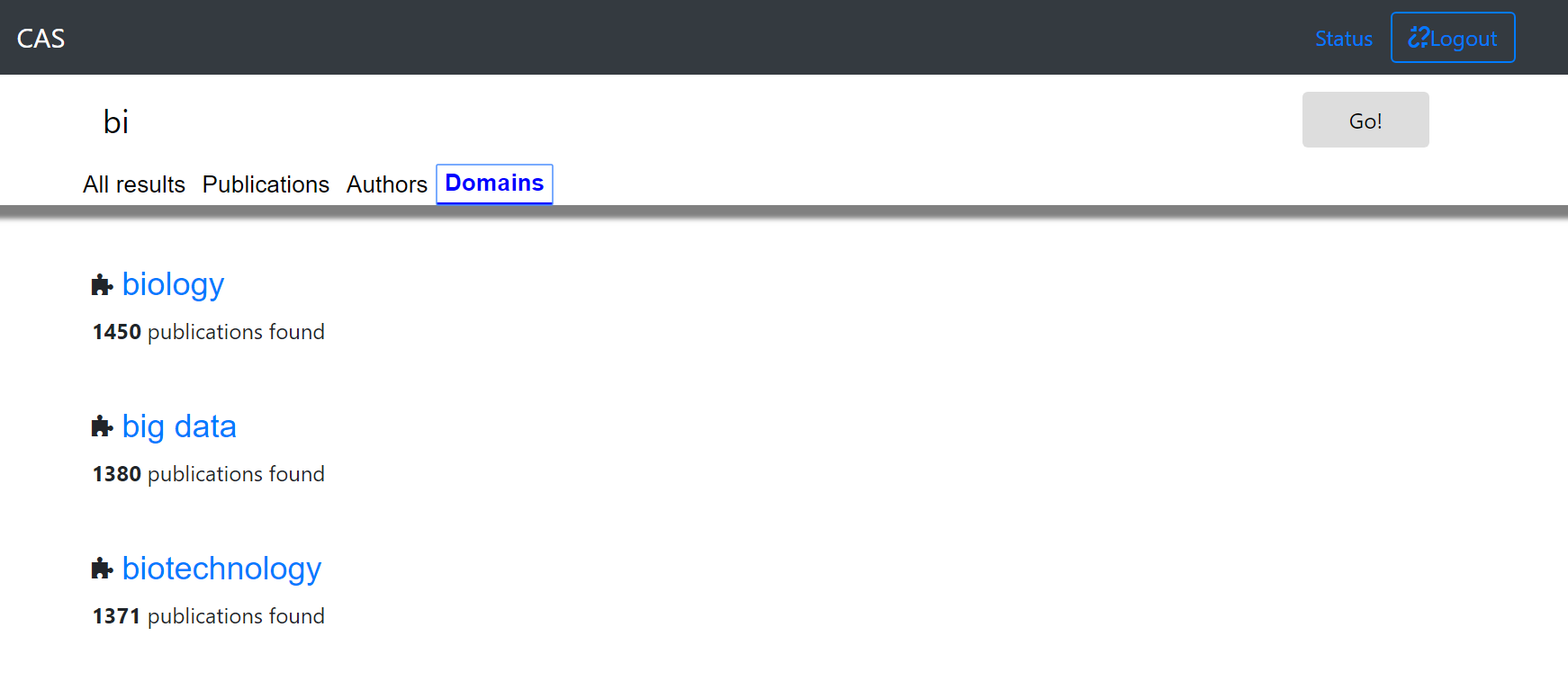


Рис 4.6 – только области знаний

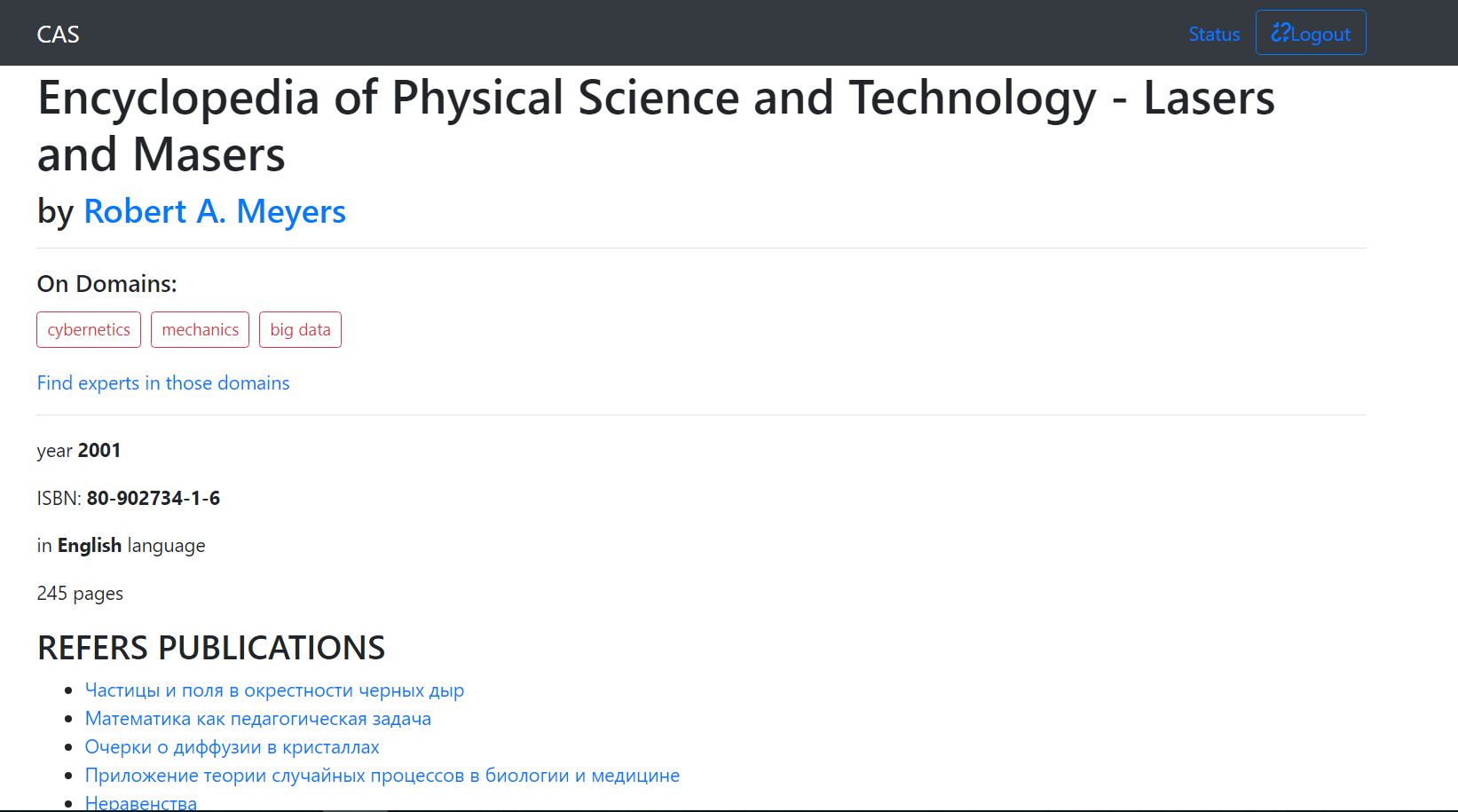
Для каждой «сущности» (автора, публикации, области знаний) создана страница подробной информации.  


Рис 4.7 – конкретная публикация

На странице публикации присутствуют ссылки на страницу информации об авторе, ссылка на поиск экспертов в тех областях, к которым относится данная публикация, список публикаций, на которые она ссылается.

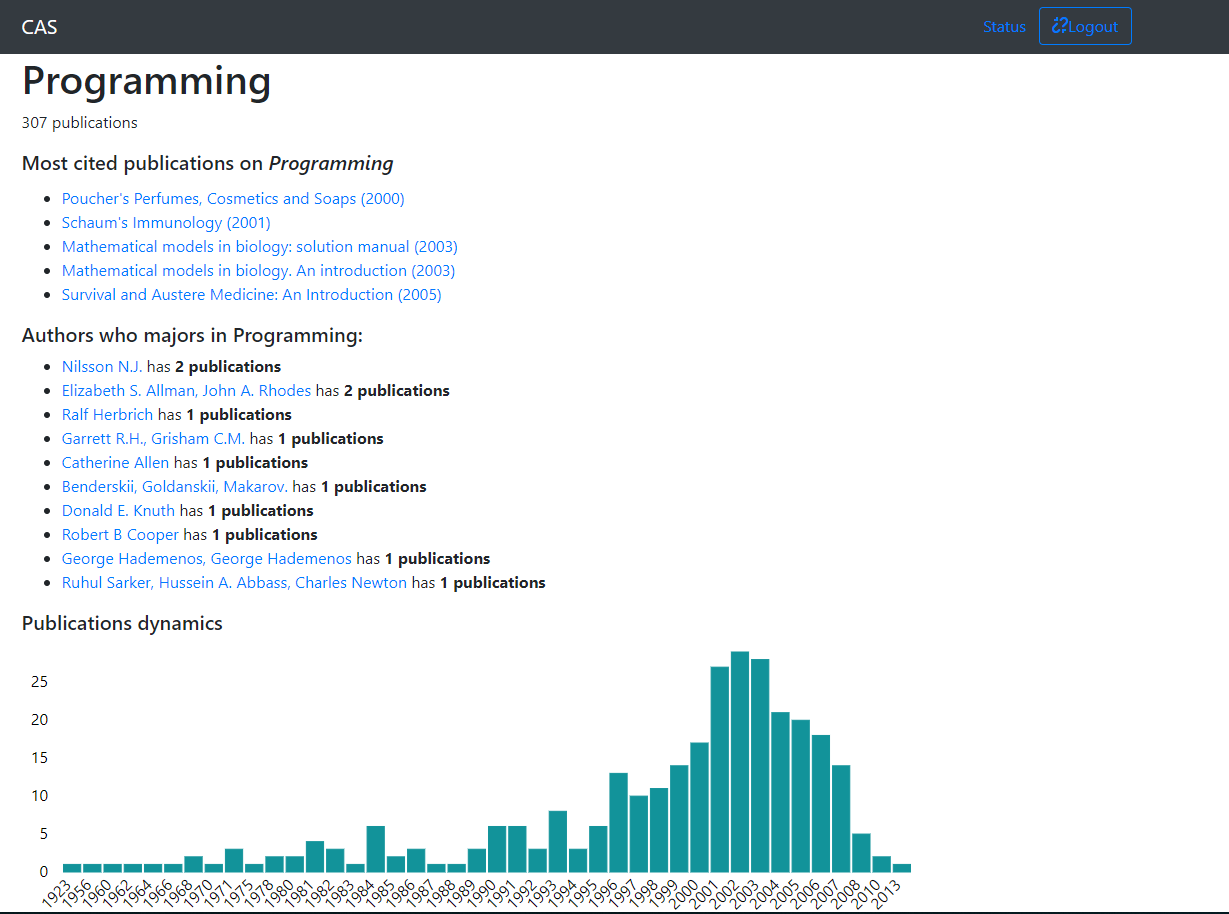
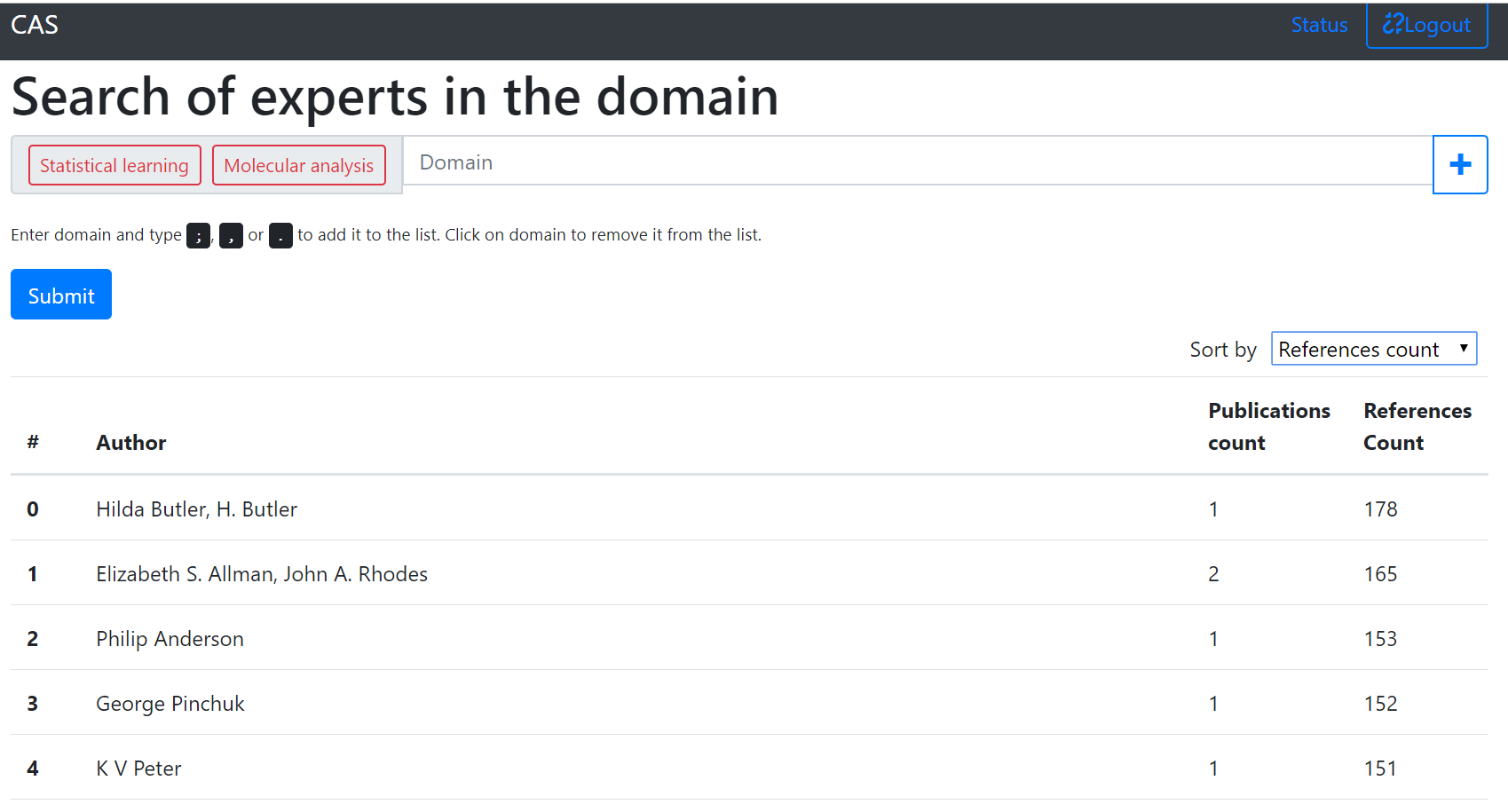


Рис 4.8 – Страница области знаний «biotechnology»

Поиск авторов-экспертов в области(областях) свои результаты выдает на основе количества публикаций у автора в обозначенных сферах науки и количестве ссылок на эти публикации. Пример выдачи – рис 4.8.

Рис 4.9 – поиск экспертов.

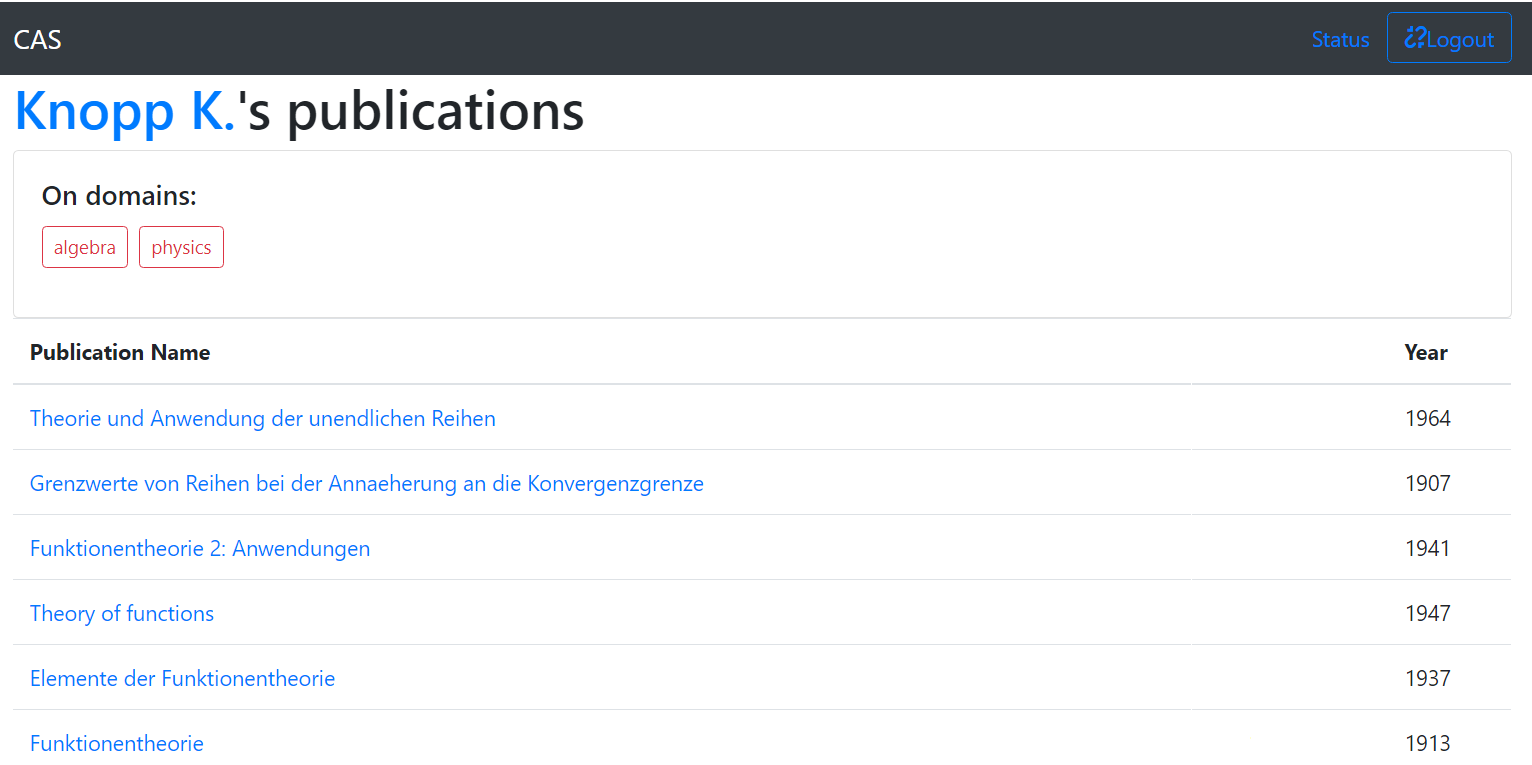


Рис 4.10 – публикации конкретного автора в обозначенных областях

Вывод областей знаний.

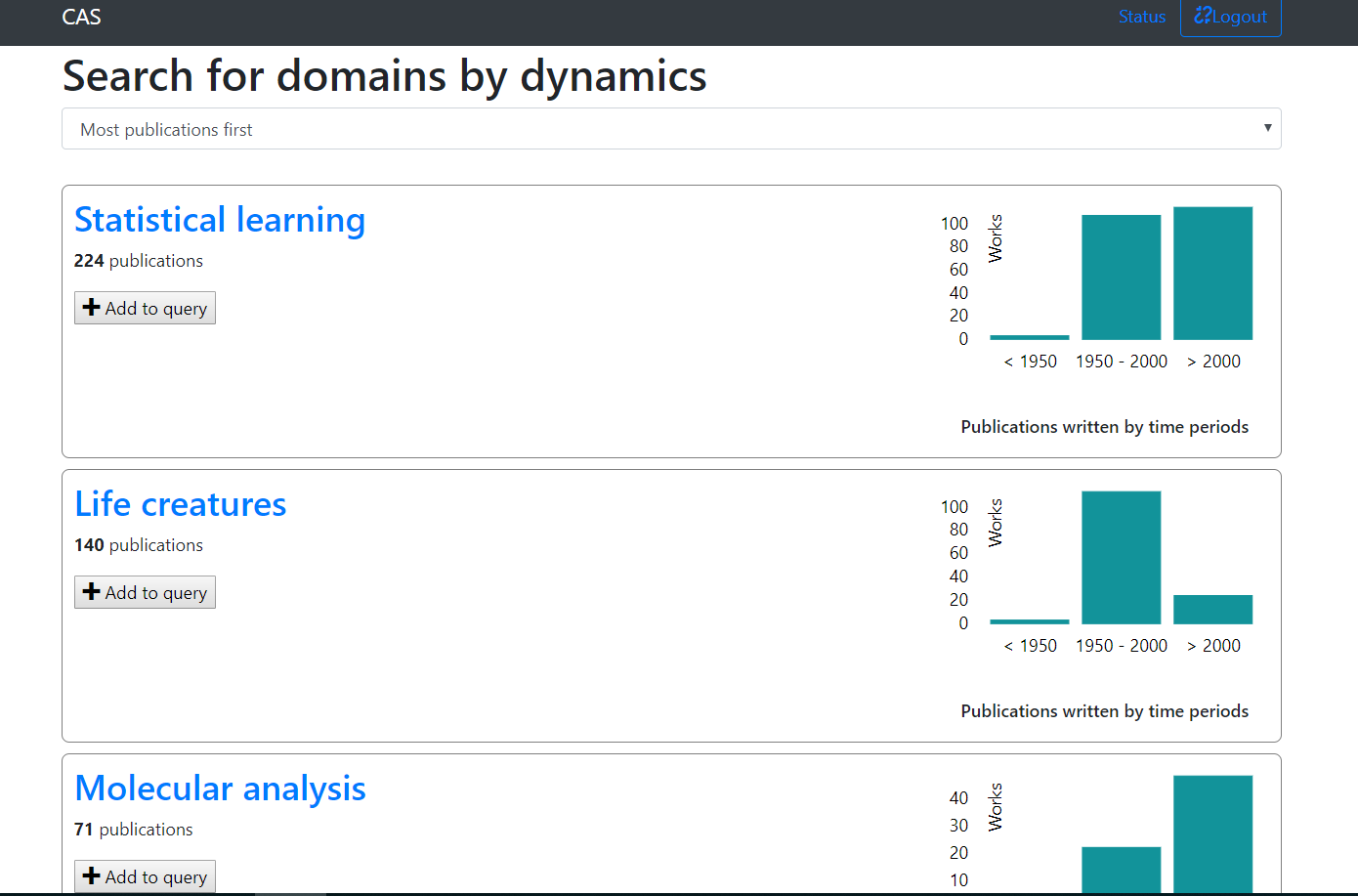
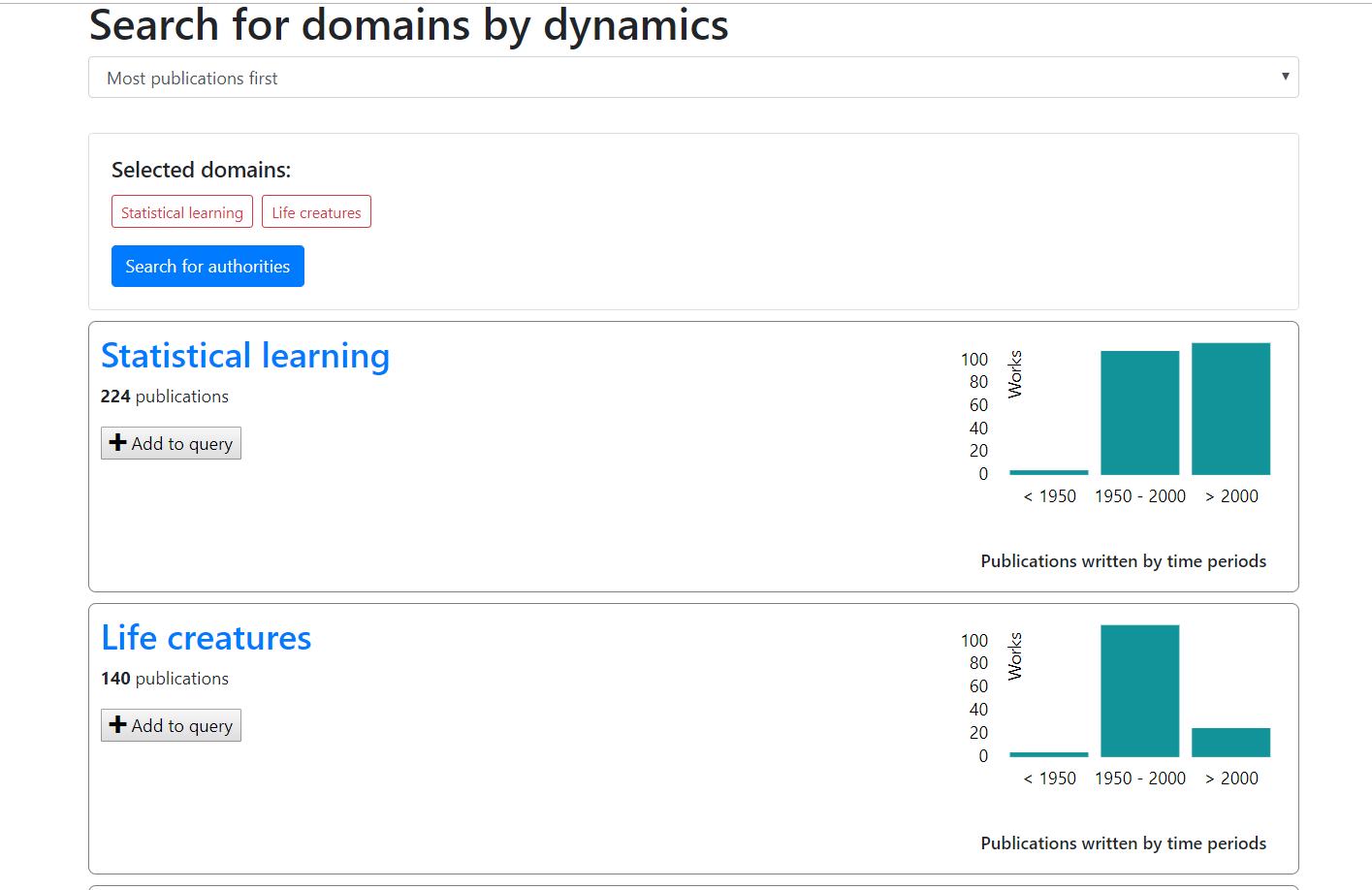


Рис 4.11 – области знаний по их популярности

Кроме просмотра списка можно, нажимая на кнопку «+» справа выбрать некоторые области и осуществить поиск экспертов в них. (Рис 4.10.).



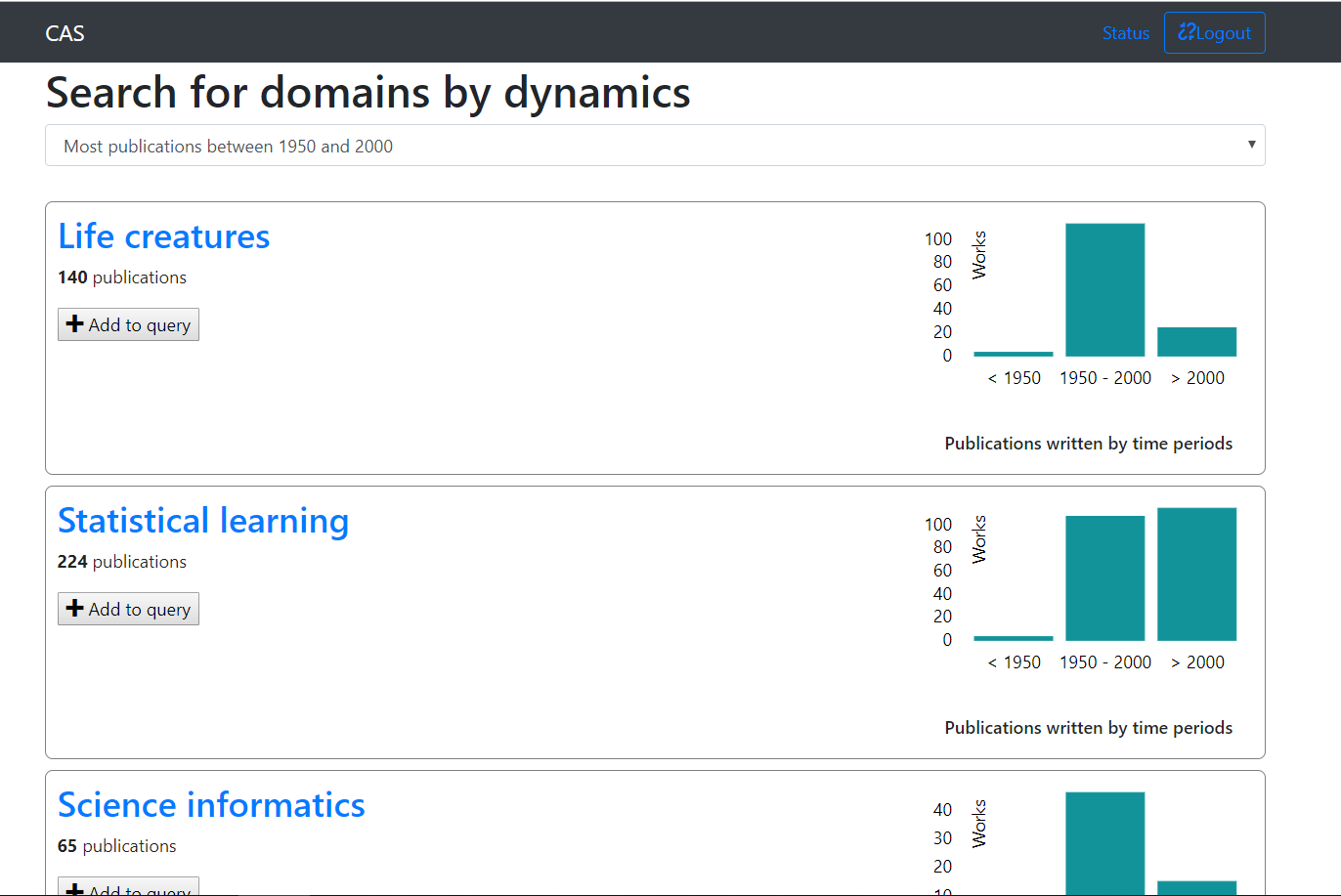
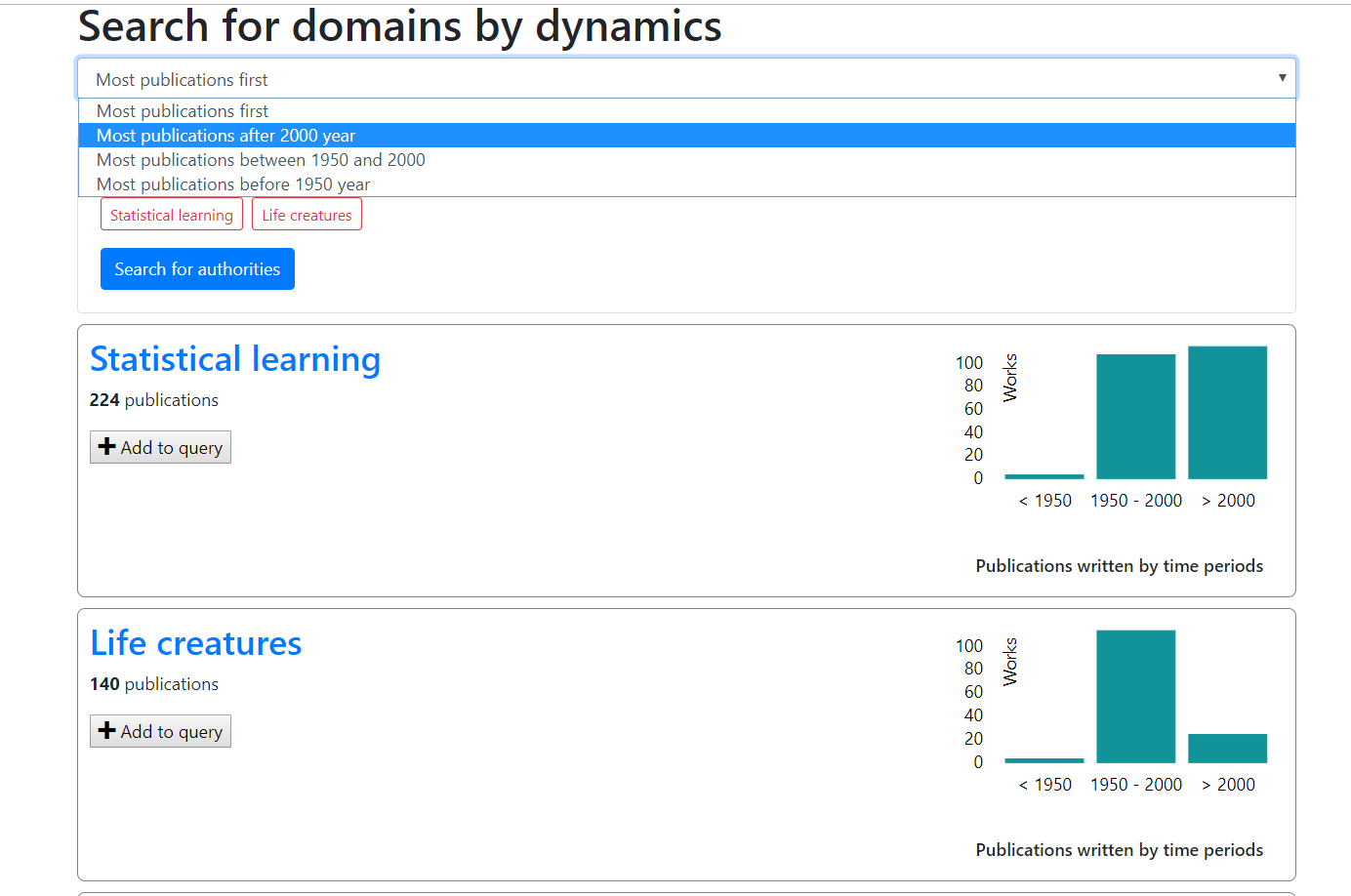


Рис 4.12



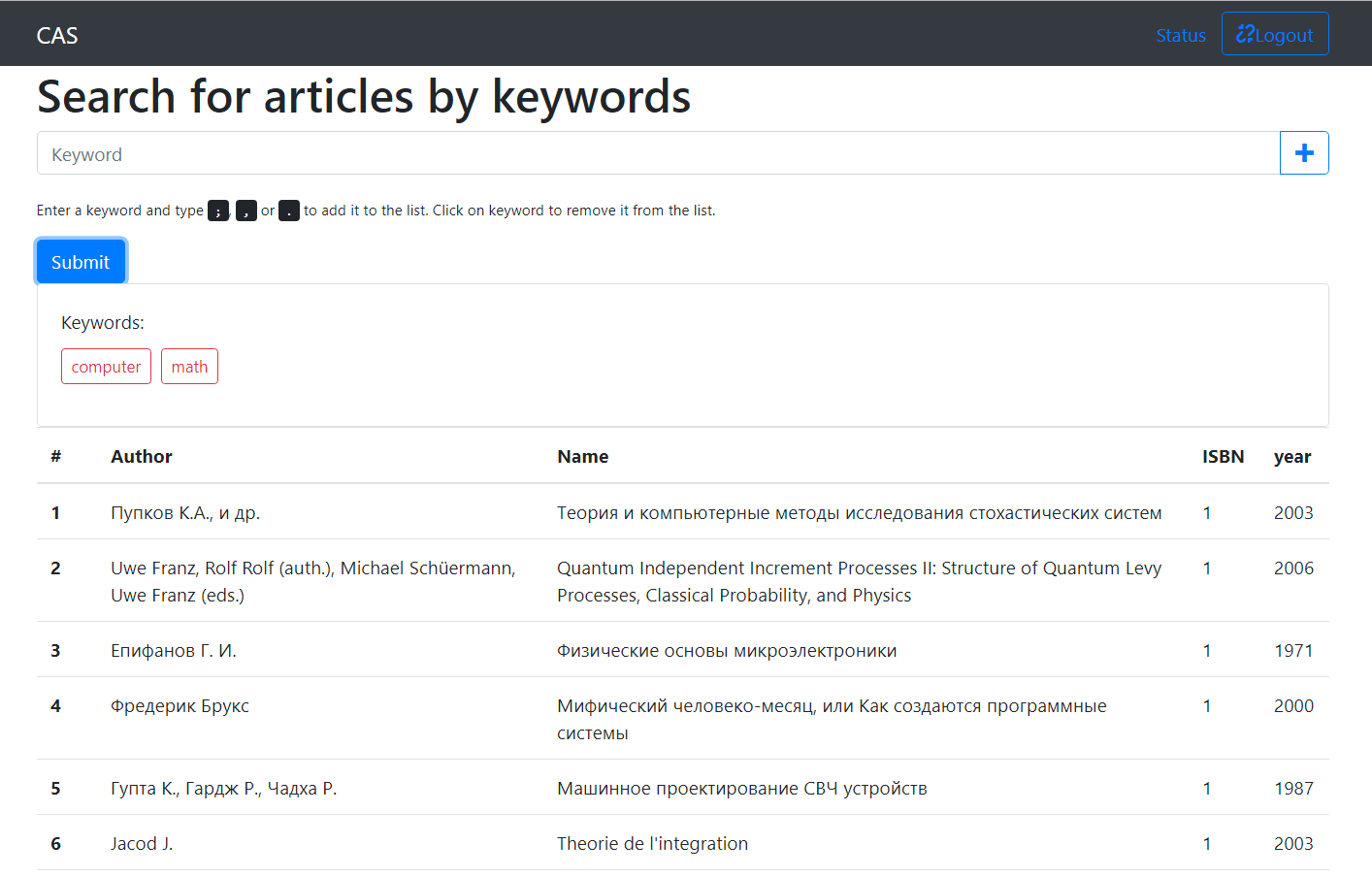


Рис 4.13 – поиск публикаций по ключевым словам

Конструктор запросов. Доступны 2 режима: поле ввода, куда можно вручную вводить запрос, и режим, который позволяет строить простые запросы при помощи дополнительных элементов интерфейса пользователя, т.е. глубоких знаний языка запросов Cypher не требуется. В случае ошибки в запросе выводится ее описание, которое предоставляет py2neo.

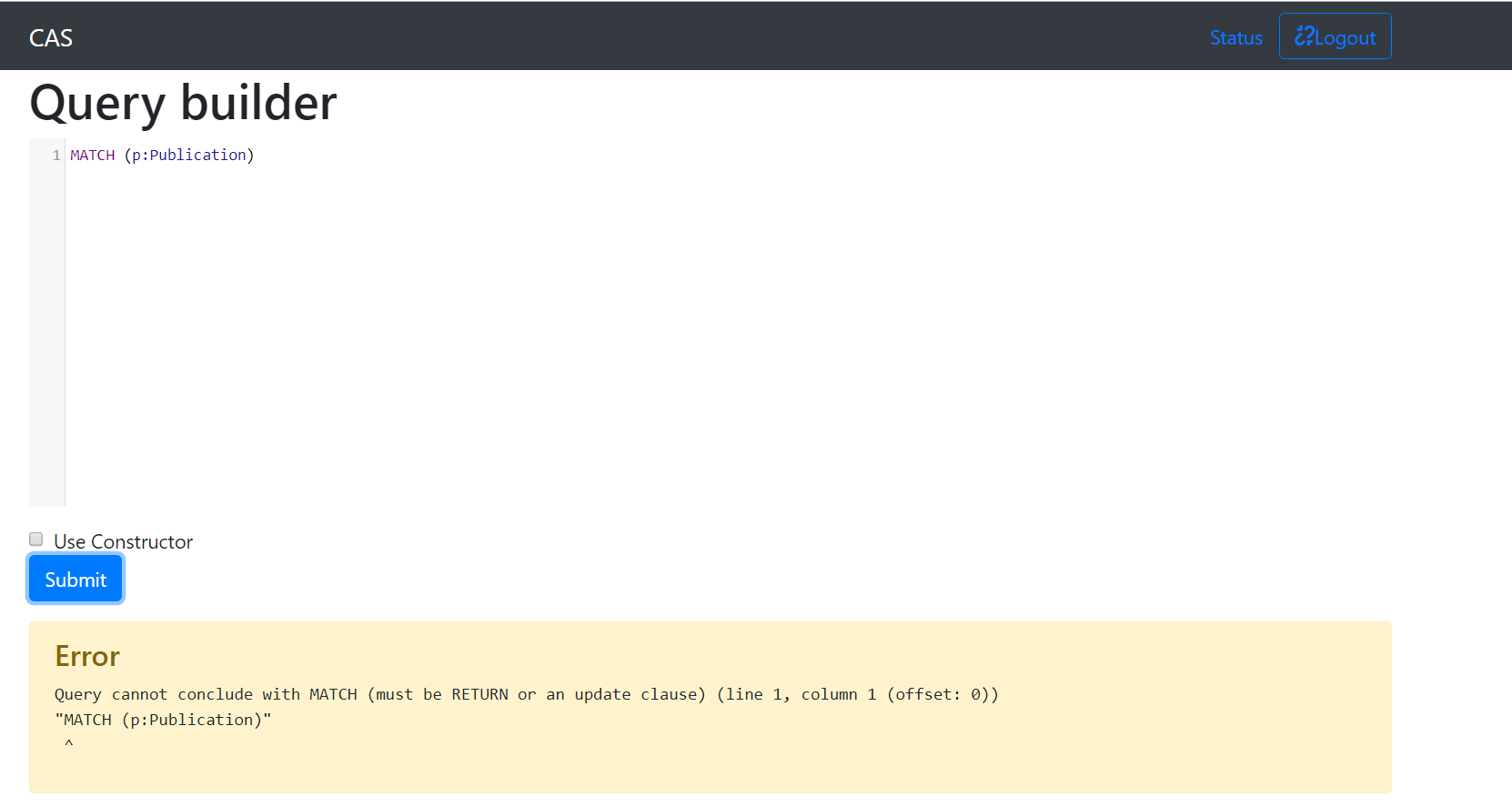


Рис 4.14 – сообщение о неправильно написанном запросе

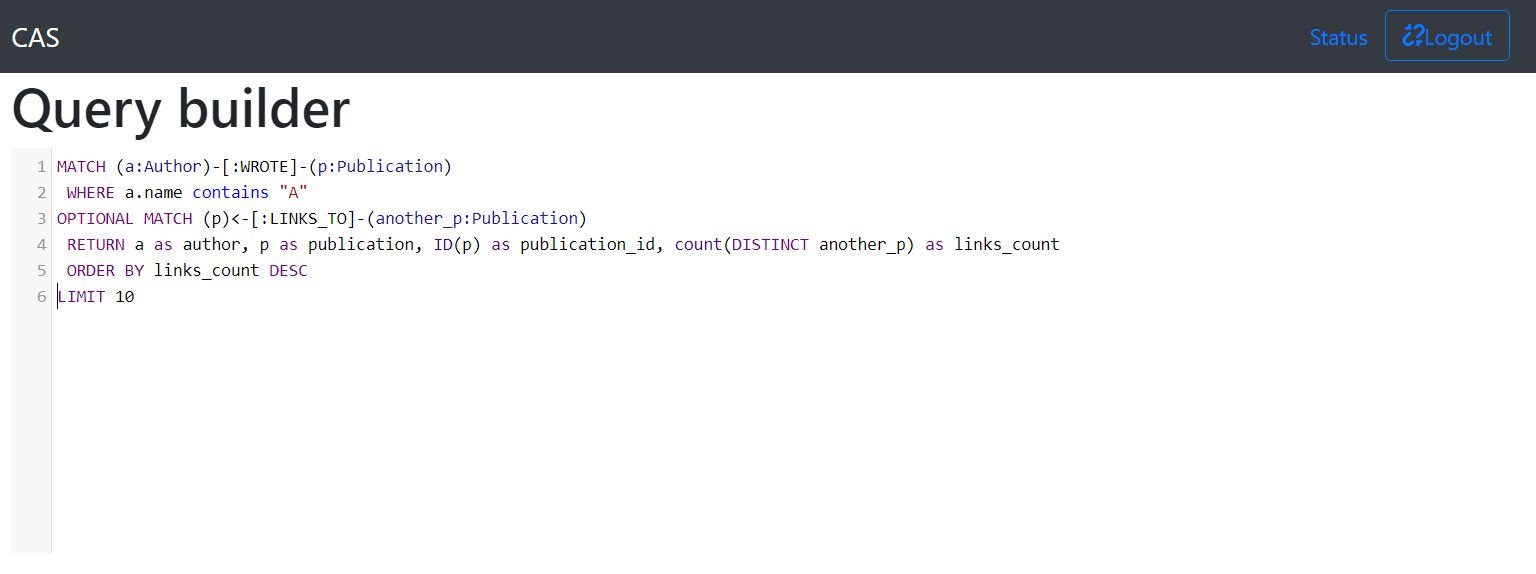
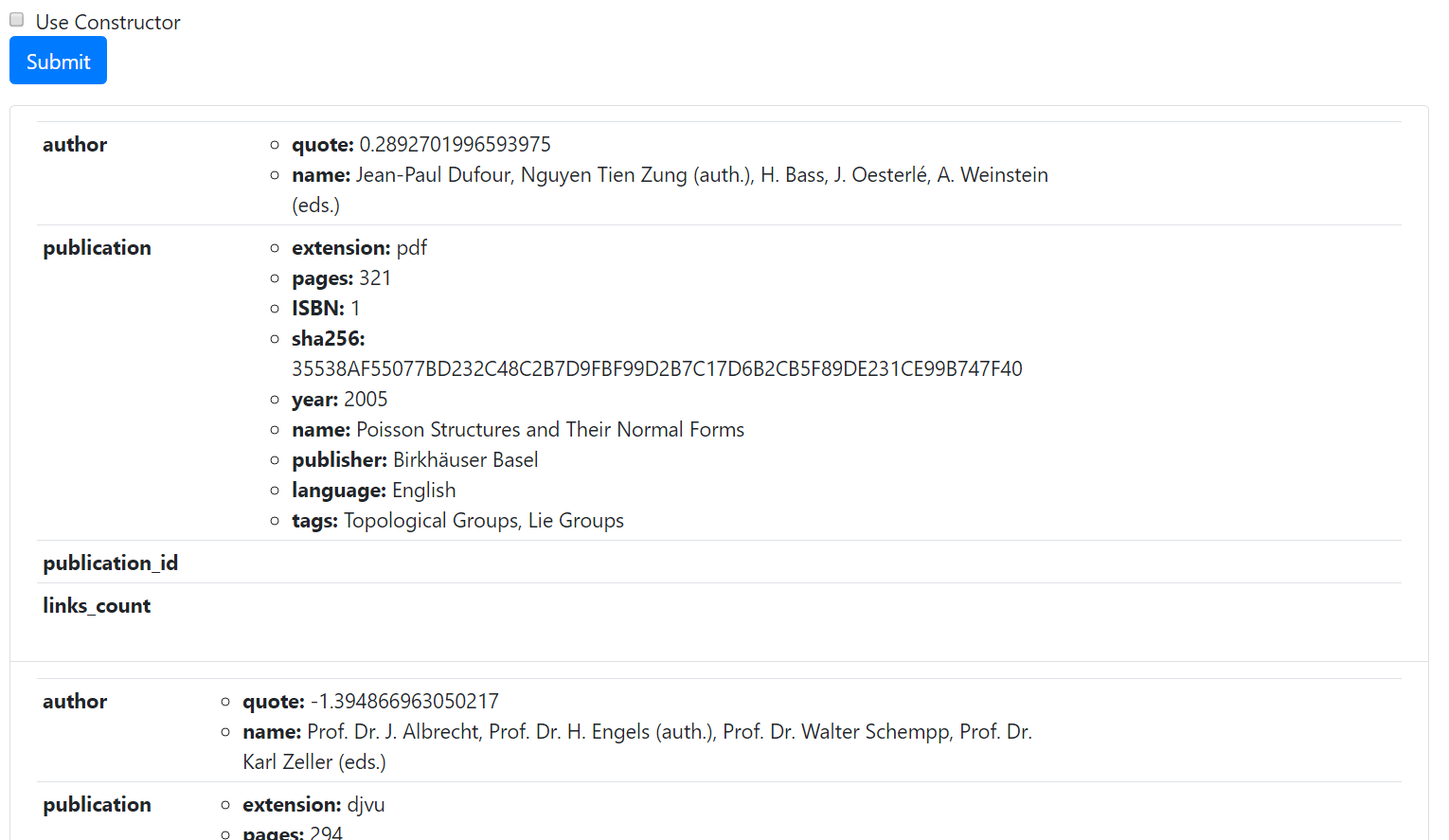
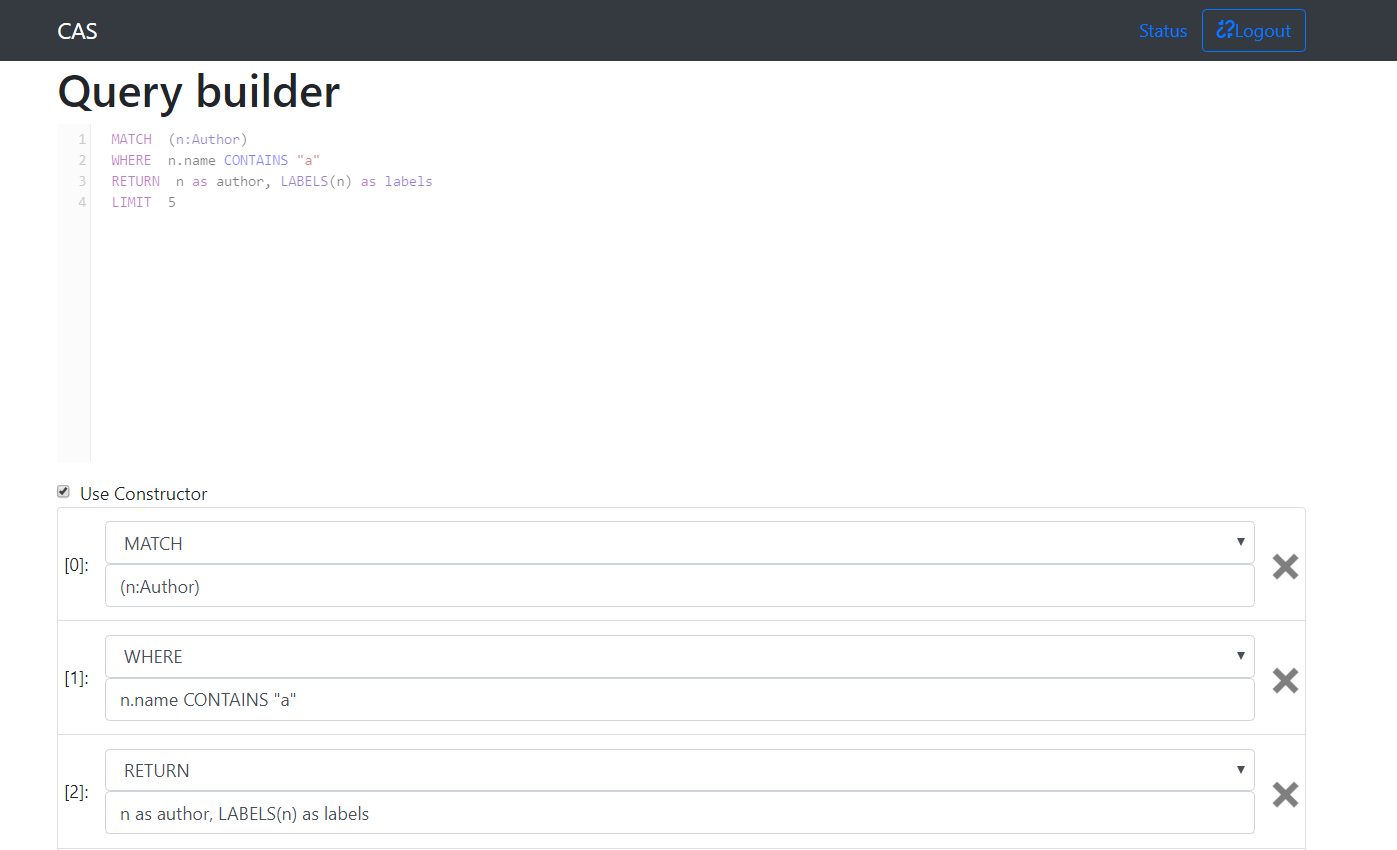
  


Рис 4.15 – запрос и его результаты



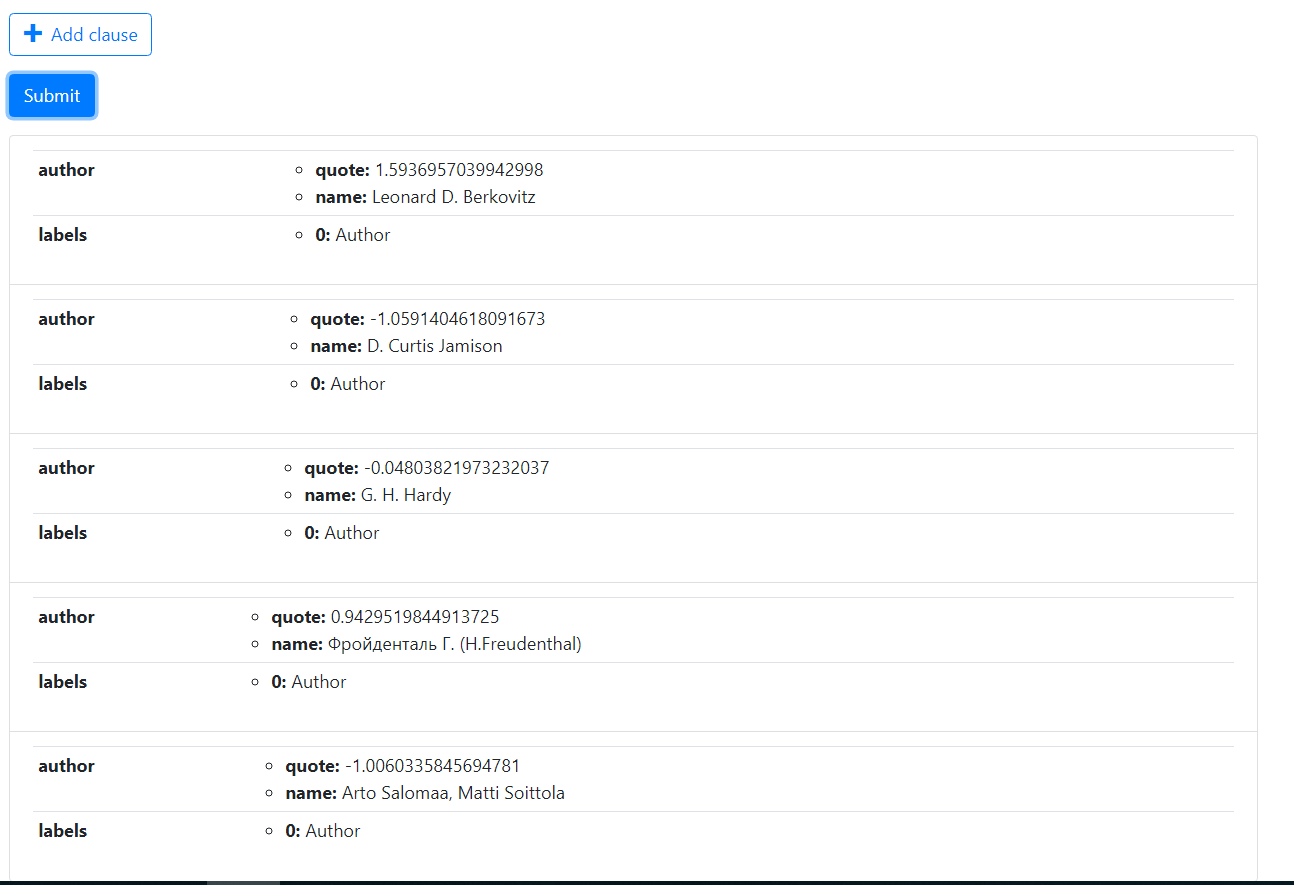


Рис 4.16 – запрос, созданный с помощью визуального конструктора и результаты

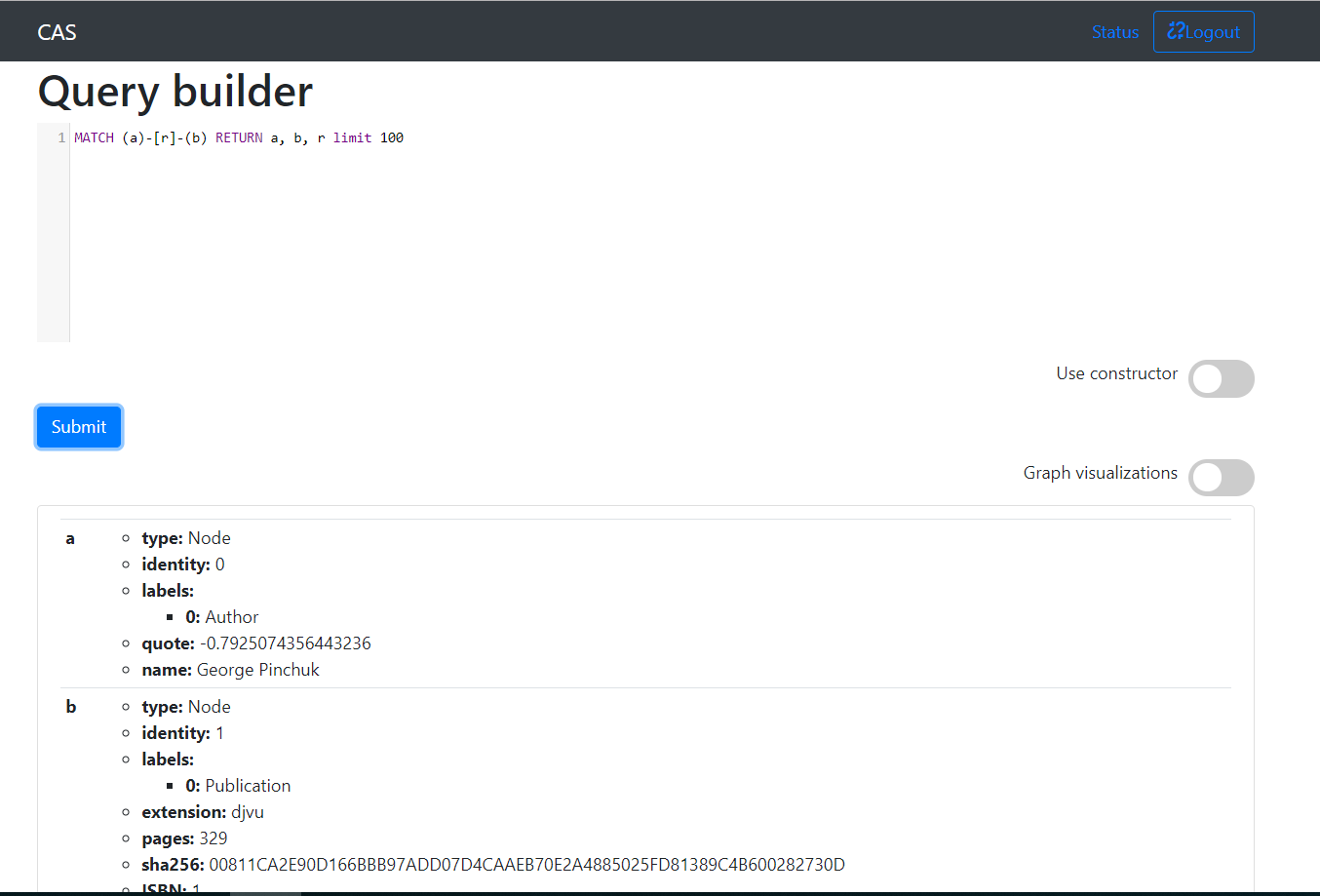


Рис 4.17 – У конструктора запросов есть 2 режима отображения

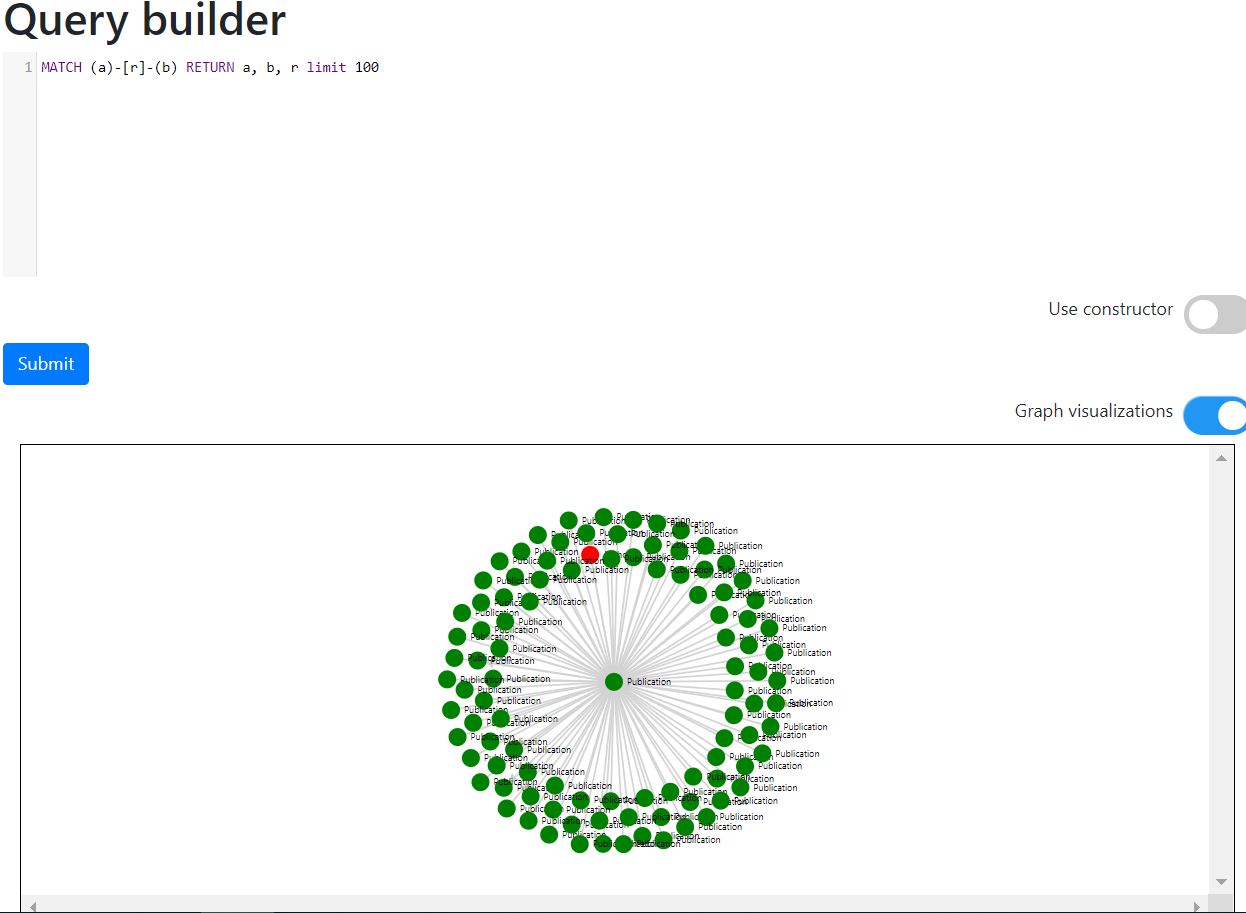


Рис 4.18 – Визуализация результата динамическим графом с помощью библиотеки react-d3-graph

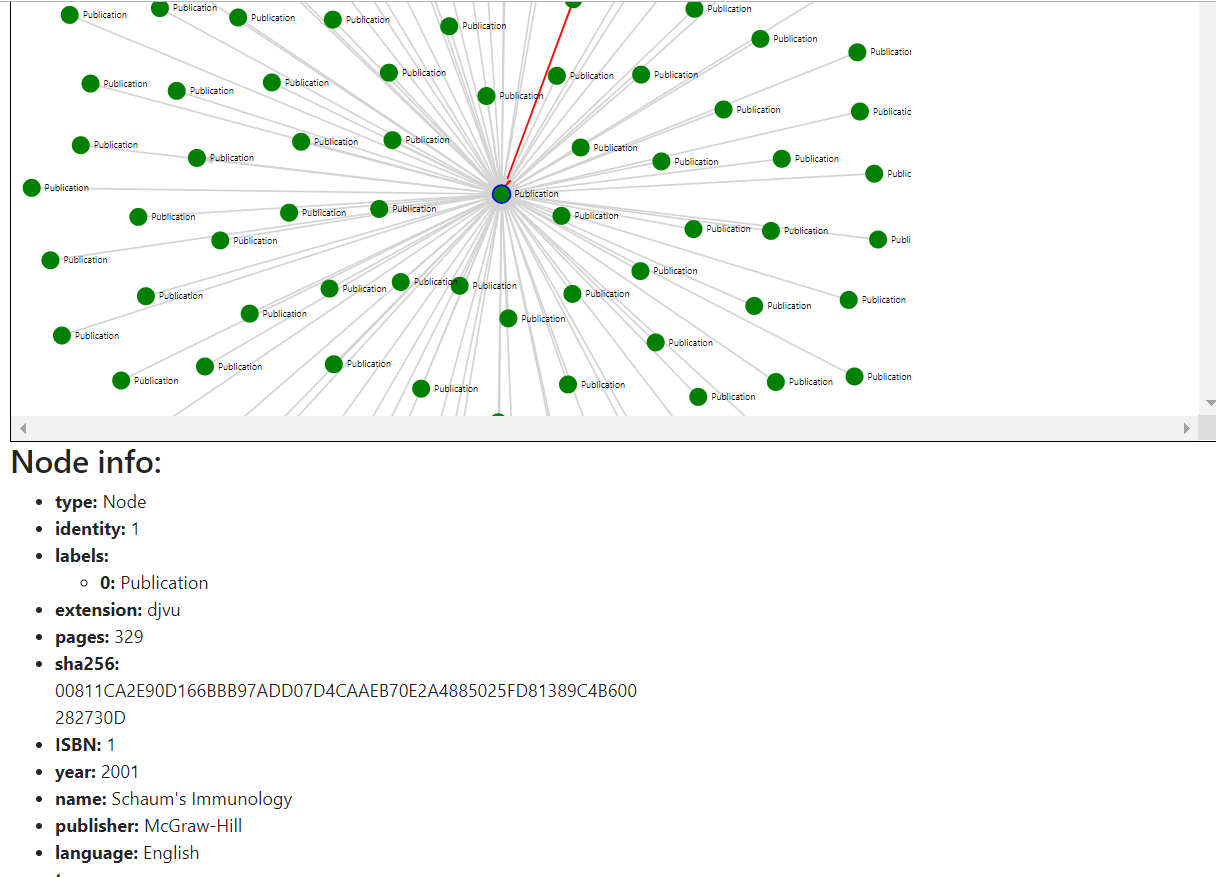


Рис 4.19 – можно просматривать свойства конкретного узла

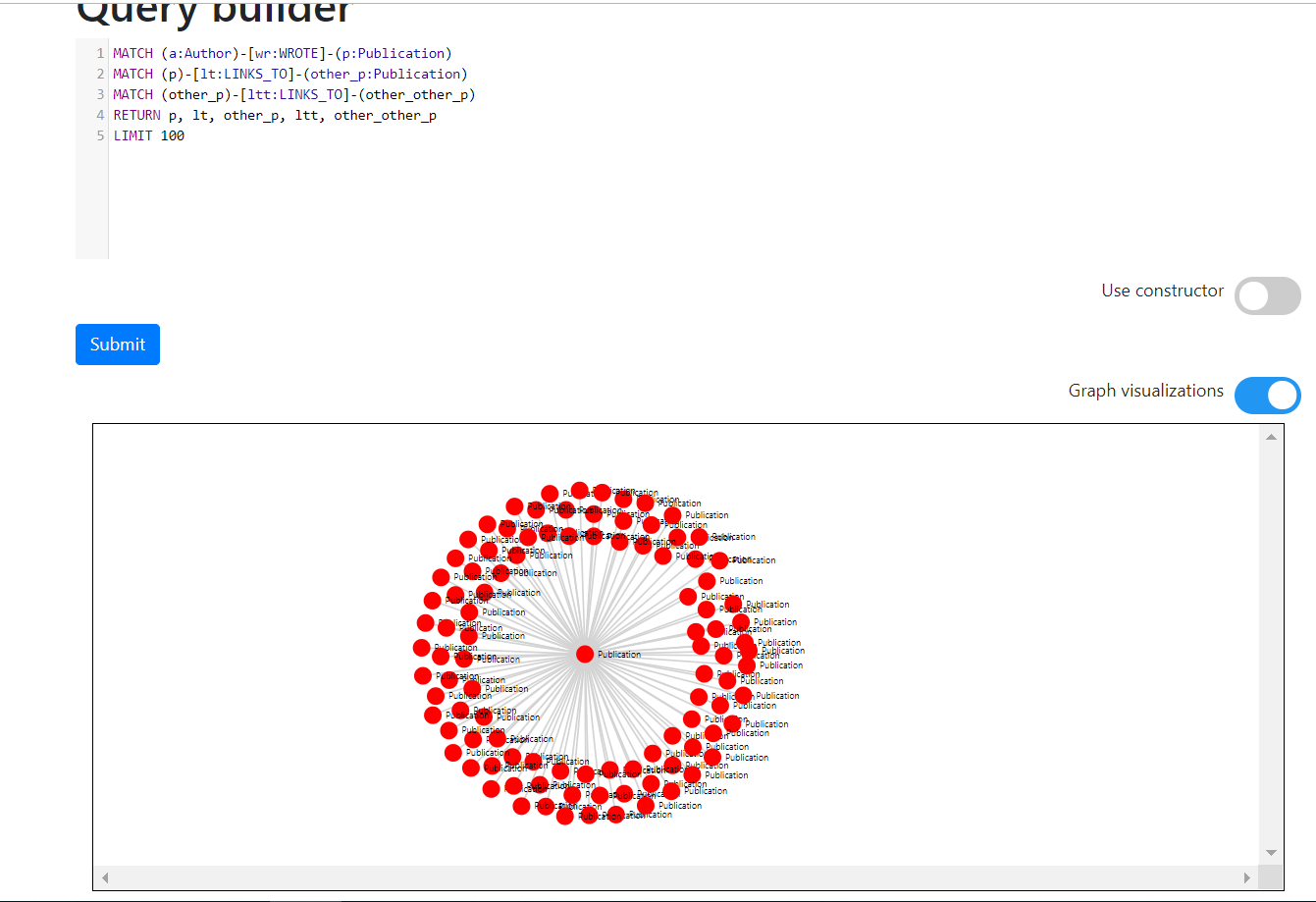


Рис 4.20 – еще выдача графа

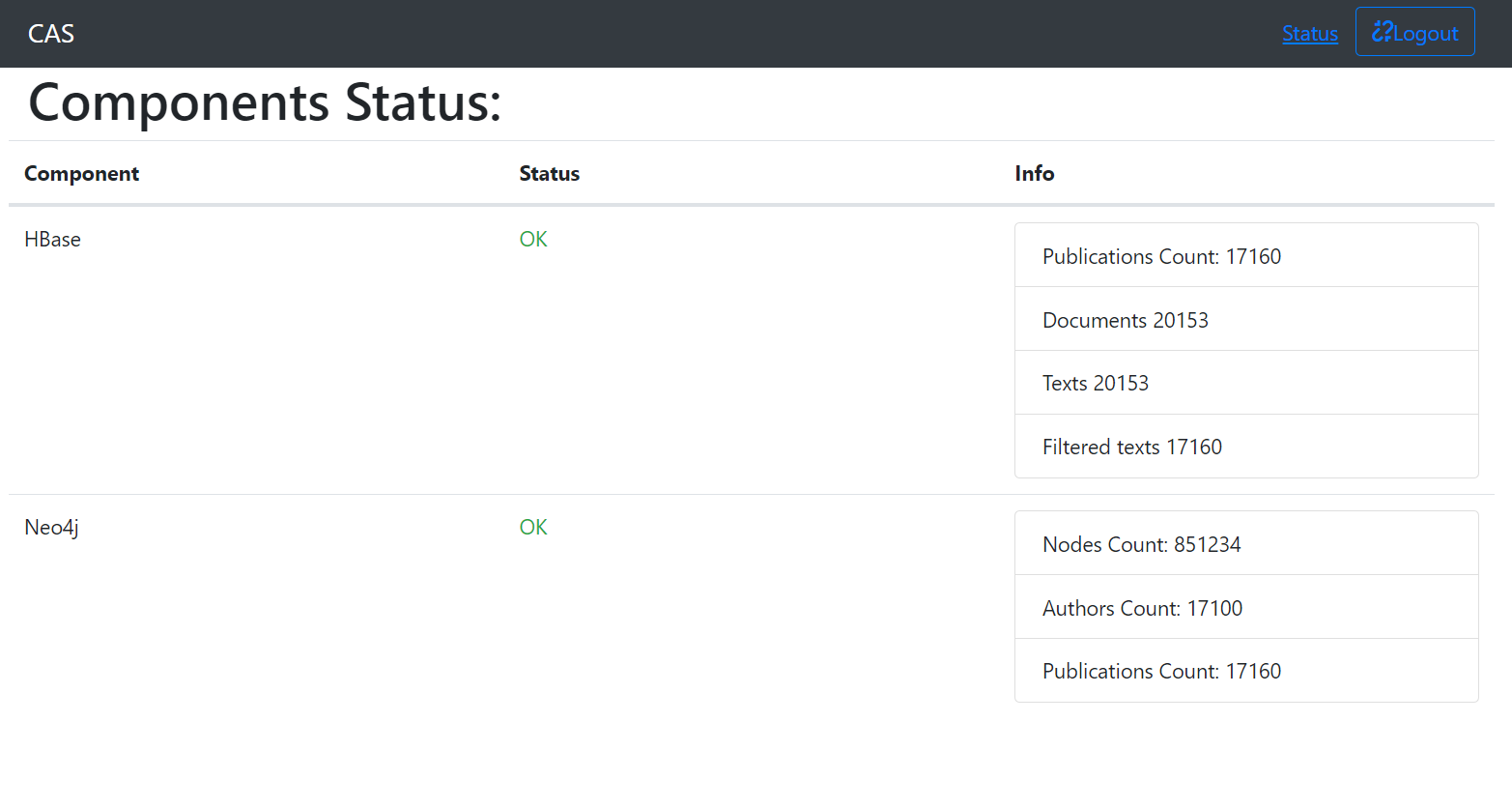


Рис. 4.21 – страница статуса баз данных