

## Вариант 1. Ремонт бытовой техники

### Elasticsearch.

#### 1. Типы документов (json):

Заказ:

```
{index, doc_type, id, body: {id_заказа, дата_заказа, id_заказчика, сведения_о_заказчике*, данные_о_заказе*, срок_выполнения_заказа, фактическая_дата_выполнения, [запчасть*], [ремонт*], стоимость, id_мастера}}
```

Мастер:

```
{index, doc_type, id, body: {сведения_о_мастере*, [отзыв_о_работе*]}}
```

Примечание. Квадратные скобки [] обозначает тег (может быть несколько значений)

#### 2. Требование к анализатору:

поля, отмеченные \*, разделить на слова, убрать пунктуацию с помощью токенизатора standart (русский), перевести все токены в нижний регистр, убрать токены, находящиеся в списке стоп-слов, выполнить стемминг оставшихся токенов с помощью фильтра snowball.

#### 3. Запросы с вложенной агрегацией:

- разбить заказы по дате заказа с периодом 1 месяц, для каждой «корзины» определить суммарное число заказов по каждой запчасти,
- вывести мастеров, в сведениях которых указан стаж работы.

### Neo4j.

1. По данным из Elasticsearch заполнить графовую базу данных  
Заказ(id\_заказа, дата\_заказа, сведения\_о\_заказчике, стоимость\_заказа) -  
Выполнил(срок\_выполнения\_заказа, фактическая\_дата\_выполнения) -  
Мастер(id\_мастера, сведения\_о\_мастере).

Примечание. В скобках приведены свойства узлов и отношения (связи), глагол – это отношение.

2. Разработать и реализовать запрос: найти мастера, который выполнил максимальное количество заказов.

### Spark

1. По данным из Elasticsearch сформировать csv-файлы (с внутренней схемой) таблиц «Заказчик», «Заказ», «Мастер» и сохранить их в файловой системе HDFS.

2. Написать запрос select: найти заказы и мастеров, которые не выполнили заказы в срок.

3. Реализовать этот запрос в Spark. Построить временную диаграмму его выполнения по результатам работы монитора.

-----