**Вариант 17. Строительная компания**

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, стоимость\_заказа, id\_бригады}}

Бригада:

{index, doc\_type, id, body: {сведения\_о\_бригаде\*, [член\_бригады], [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 год, для каждой группы определить суммарную стоимость заказов по каждой бригаде,

- предложить признаки отрицательного отзыва; определить бригады хотя бы с одним отрицательным отзывом.

Neo4j.

1. По данным из Elasticsearch заполнить графовую базу данных Заказ(id\_заказа, дата\_заказа, сведения\_о\_заказчике, стоимость\_заказа) - Выполнила(срок\_выполнения\_заказа, фактическая\_дата\_выполнения) - Бригада(id\_бригады, сведения\_о\_бригаде).

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

2. Разработать и реализовать запрос: найти заказы и бригады, которые выполнили заказы с превышением срока.

Spark

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

2. Написать запрос select: найти суммарную стоимость заказов по каждому заказчику.

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

-----------------------------------------------------