

# Neo4j

## Граф собственных свойств

### point

id(Integer) - идентификатор точки из openstreetmap

lat(Float) - широта точки

lon(Float) - долгота точки

street(String) - название улицы

### relations

id(Integer) - идентификатор отношения

e.id(Integer) - идентификатор начального узла

d.id(Integer) - идентификатор конечного узла

### road\_section

id(Integer) - идентификатор секции

lat(Float) - широта начальной точки

lon(Float) - широта конечной точки

### plan

id(Integer) - идентификатор плана ремонта

name(String) - название плана

## Размер данных

$$V_{\text{point}} = V_{\text{id}} + V_{\text{lon}} + V_{\text{lat}} + V_{\text{street}} = 8\text{B} + 4\text{B} + 4\text{B} + 32\text{B} = 48\text{B}$$

$$V_{\text{relations}} = V_{\text{id}} + V_{\text{e.id}} + V_{\text{d.id}} = 8\text{B} + 8\text{B} + 8\text{B} = 24\text{B}$$

$$V_{\text{road\_section}} = V_{\text{id}} + V_{\text{lat}} + V_{\text{lon}} = 8\text{B} + 4\text{B} + 4\text{B} = 16\text{B}$$

$$V_{\text{plan}} = V_{\text{id}} + V_{\text{name}} = 8\text{B} + 16\text{B} = 24\text{B}$$

Размер участка дороги сильно зависит от количества точек, принадлежащих ей, поэтому возьмем среднее значение в 3 точки. Примем среднее количество участков дороги в одном плане равным 2, тогда:

$$V1(\text{Объем данных плановых работ}) = V_{\text{road\_section}} + V_{\text{plan}} + (\sim 4) * V_{\text{relations}} = 16B + 24B + (\sim 4) * 24B = 136B$$

$$V2(\text{Объем данных хранимых узлов и отношений}) = V_{\text{point}} + (\sim 2) * V_{\text{relations}} = 48B + (\sim 2) * 24B = 96B$$

$$V = V2 + V1 = 136B + 96B = 232B$$

Размер используемой памяти зависит от количества планов и количества узлов, следовательно

$Vp(N) = N * V1 = N * 136B$  - объем памяти для плановых работ, где N - количество планов ремонтных работ

$Vg(N) = N * V2 = N * 96B$  - объем памяти хранимых узлов и отношений, где N - количество хранимых узлов

## Избыточность модели

Фактический объем  $V1 = 232B$

Чистый объем  $V2 = 232B - (\sim 2) * 24B = 232B - 48B = 184B$

Избыточность данных -  $V1 / V2 = 232B / 184B = 1.26$

## Примеры запросов, реализующих сценарий использования

Получение все планы работ и упорядочим их по названию:

```
MATCH (ee:Plan) RETURN ee.name ORDER BY ee.name
```

Добавление связи между двумя точками карты:

```
MATCH (e:Point) WHERE e.id = 123
```

```
MATCH (d:Point) WHERE d.id = 124
```

```
CREATE (e)-[:connects]->(d), (d)-[:connects]->(e)
```

## Сравнение моделей

- NoSQL модель данных занимает меньше данных по объему

- Большое количество различных таблиц в SQL модели
- NoSQL модель является более оптимальной с точки зрения сложности и объема выполняемых запросов
- Neo4j модель является более наглядной с точки зрения графового представления сущностей и связей

## Выводы

NoSQL является наиболее ёмкой моделью по объему занимаемой памяти и более оптимальной с точки зрения сложности и объема выполняемых запросов. Однако Neo4j позволяет получить наглядное представление модели в виде графа, а SQL в виде таблиц.