

图数据库Neo4j

给节点的某个属性添加唯一性约束

```
CREATE CONSTRAINT ON (n :Gakkiyomi) ASSERT n.name IS UNIQUE;
```

根据两个已有节点创建一条关系并返回

```
MATCH (a),(b) WHERE a.name="fangc" AND b.name="ssg229" CREATE (a) -[x:Layer3 {vlan: 131}]> (b) return x;
```

节点添加新属性

```
MATCH (n:Gakkiyomi) WHERE n.id = "a22" SET n.sex = "男" RETURN n
```

删除指定关系

```
MATCH (p1:Gakkiyomi)-[r:Friend {id:"349f8fa3-65b7-4c53-b047-1d6c3aa5ec8c"}]-(p2:Firewall)
DELETE r
```

修改节点label

```
match(n:oldlabel) set n:newlabel remove n:oldlabel
```

修改关系label

```
MATCH p=(n:User)-[r:subord]->(e:department) create(n)-[r2:subord_new]->(e)
set r2 = r with r delete r
```

根据默认生成id删除节点

```
MATCH (r),(b) WHERE id(r) = 10 AND id(b) = 9 Delete r,b
```

查询一条关系(返回 节点-关系-节点)

```
MATCH p=()-[r:Friend {id:"2ec0ddde-0eb1-4f6b-a9e4-094cfbdfc694"}]->() RETURN p
```

查询一条关系(只返回关系)

```
MATCH p=()-[r:Friend ]->() RETURN r
```

查询一条关系(返回关系和节点label)

```
MATCH p=(a)-[r:Friend]->(b) with p as ps, labels(a) as x, labels(b) as y return ps,x,y
```

查询label名

```
MATCH (r:Firewall) RETURN distinct labels(r)
```

查询两点之间的最短路径 (3 为在路径深度为3以内中查找)

```
match(n{name:"哈士奇"},(m{name:"fangc"}))with n,m match p=shortestpath((n)-[*]->(m)) return p;
```

```
match(n{name:"哈士奇"},(m{name:"ssg229"}))with n,m match p=shortestpath((n)-[*..3]-(m)) return p;
```

shortestpath 查询一条

allshortestpath 查询所有

查询两点之间的所有路径

```
MATCH p=(a)-[*]->(b)
RETURN p
```

查询数组里的属性 [1,2,4,5]

```
match (n) where 5 in n.ip return n
```

修改节点属性

```
MATCH (a:Ta{names:"afaf"})
SET a.names="a"
return a
```

修改节点属性名称

```
match(n) set n.propertyNew=n.propertyOld remove n.propertyOld
```

查询多label多条件

```
match (n) where any(label in labels(n) WHERE label in
['HDSStorage','BrocadePort']) and '192.168.1.106' in n.ip or n.domain = '28'
return n
```

Cypher语句规则和具备的能力:

- Cypher通过模式匹配图数据库中的节点和关系，来提取信息或者修改数据。
- Cypher语句中允许使用变量，用来表示命名、绑定元素和参数。
- Cypher语句可以对节点、关系、标签和属性进行创建、更新和删除操作。
- Cypher语句可以管理索引和约束。

运算符

常规运算	DISTINCT, ., []
算数运算	+, -, *, /, %, ^
比较运算	=, <>, <, >, <=, >=, IS NULL, IS NOT NULL
逻辑运算	AND, OR, XOR, NOT
字符串操作	+
List操作	+, IN, [x], [x .. y]
正则操作	=~
字符串匹配	STARTS WITH, ENDS WITH, CONTAINS

变长路径检索

- 变长路径的表示方式是：[*N...M]，N和M表示路径长度的最小值和最大值。
- (a)-[*2]->(b)：表示路径长度为2，起始节点是a，终止节点是b；
 - (a)-[*3...5]->(b)：表示路径长度的最小值是3，最大值是5，起始节点是a，终止节点是b；
 - (a)-[*...5]->(b)：表示路径长度的最大值是5，起始节点是a，终止节点是b；
 - (a)-[*3...]->(b)：表示路径长度的最小值是3，起始节点是a，终止节点是b；
 - (a)-[*]->(b)：表示不限制路径长度，起始节点是a，终止节点是b；

查询所有节点的属性

```
match (n) return distinct keys(n)
```

neo4j 数据导入

	create 语句	load csv 语句	Batch Inserter	Batch Import	neo4j-import
适用 场景	1~1w nodes	1w~10w nodes	千万以上 nodes	千万以上 nodes	千万以上 nodes
速度	很慢 (1000 nodes/s)	一般 (5000 nodes/s)	非常快(数万 nodes/s)	非常快(数万nodes/s)	非常快(数万nodes/s)
优点	使用方便，可实时插入。	使用方便，可以加载本地	远程CSV；可实时插入	基于Batch Inserter，可以直接运行编译好的jar包；可以在已存在的数据库中导入数据	官方出品，比Batch Import占用更少的资源
缺点	速度慢	需要将数据转换成csv	需要转成CSV；只能在JAVA中使用；且插入时必须停止neo4j	需要转成CSV；必须停止neo4j	需要转成CSV；必须停止neo4j；只能生成新的数据库，而不能在已存在的数据库中插入数据