

# Neo4j 笔记

查询所有节点

```
match (n) return n
```

创建节点

```
create (s:Scholar{name:'李峰',age:18}) return s
```

添加关系

```
match (a:Scholar),(b:Scholar) where a.name='李峰' and b.name='赵东元' create (a)-[c:cooperates{num:5}]->(b)
```

```
match (a:Scholar),(b:Scholar) where a.name='李峰' and b.name='赵东元' create (a)-[c:cooperates{num:5}]->(b)
```

查看关系

```
match p=(:Scholar{name:'李峰'})-[]-(:Scholar{name:'赵东元'})  
return p
```

修改属性

```
match (s:Scholar{name:'李峰'}) set s.age = 24 return s
```

删除节点及关系

```
match(s:Scholar{name:'李峰'}) detach delete s
```

## 复杂查询

查询赵东元所有指纹

```
match p=(:Scholar{scholarId:2019270})-[:has]->(:Stkos) return p
```

查询赵东元使用次数最高的前 6 个指纹

```
match (r:Scholar{scholarId:2019270})-[:has]->(s:Stkos) return s
order by r.num desc limit 6
```

查询使用过赵东元指纹的学者并按照 hindex 降序排序(不去重)

```
match(:Scholar{scholarId:2019270})-[:has]->(t)<-[:has]-(s)
return s.name order by s.hindex desc
```

查询使用过赵东元指纹的学者并按照 hindex 降序排序(去重)

```
match (r:Scholar{scholarId:2019270})-[:has]->()<-[:has]-(s) with
s ,count(s) as c return s.name order by s.hindex desc
```

查找使用过赵东元指纹的学者并根据匹配指纹个数降序排序

```
match (:Scholar{scholarId:2019270})-[:has]->(k)<-[:has]-(s) with  
s,count(k) as c return s.name order by c desc
```

查找使用过赵东元使用频次前六个的指纹的学者并根据匹配指纹个数降序排序

```
match (s1:Scholar{scholarId:2019270})-[r1:has]->(k) with s1,k order  
by r1.num desc limit 6 match (s1)-[:has]->(k)<-[:has]-(s2:Scholar)  
with s2,count(k) as c return s2.name,c order by c desc
```

六度空间理论（最短路径）

```
match (a:Scholar{name:'赵东元'}),(b:Scholar{name:'张海君'  
'}),p=shortestPath((a)-[*]-(b)) where all(r in rels(p) where type(r)  
='has') return p
```

