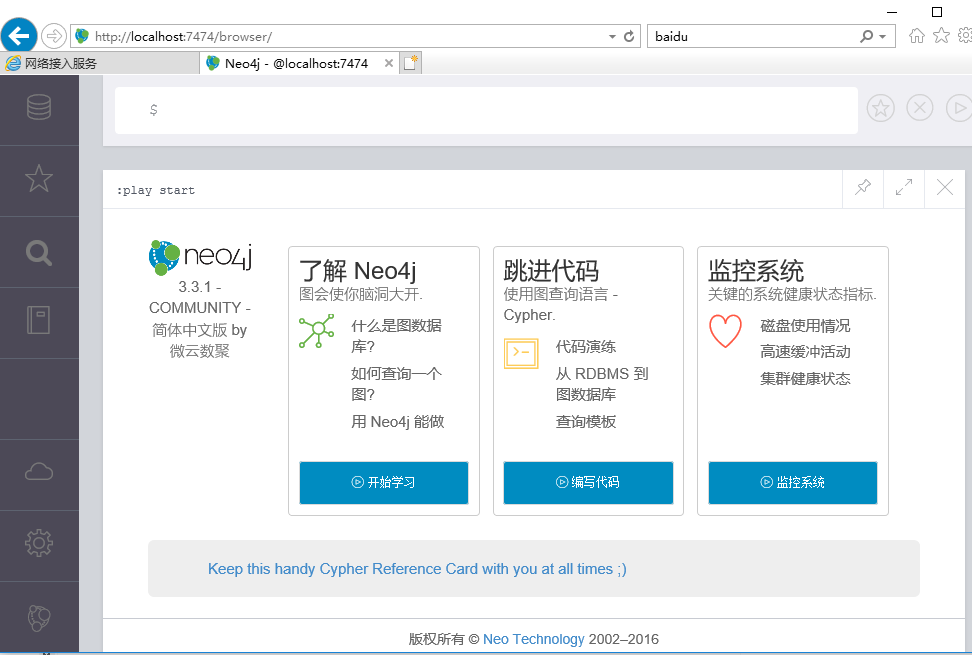
1. **Neo4j新建节点指定节点图片**

启动neo4j start, 输入localhost：7474进入浏览器中文界面。



Cypher语句如下：

CREATE

(张老师:写作成员{名称:"张老师",image:"image\作者\ZHANG.jpg"}),

(苏博士:写作成员{名称:"苏博士",image:"image\作者\SU.jpg"}),

(国防科大:大学{名称:"国防科大",image:"image\大学\GFKD.png"}),

(权威指南:著作{名称:"Neo4j权威指南",image:"image\著作\著作权威指南.png"}),

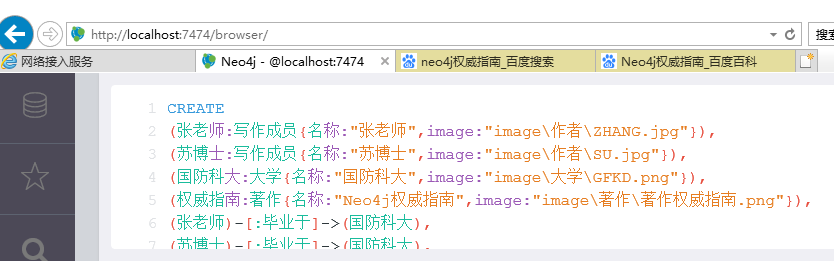
(张老师)-[:毕业于]->(国防科大),

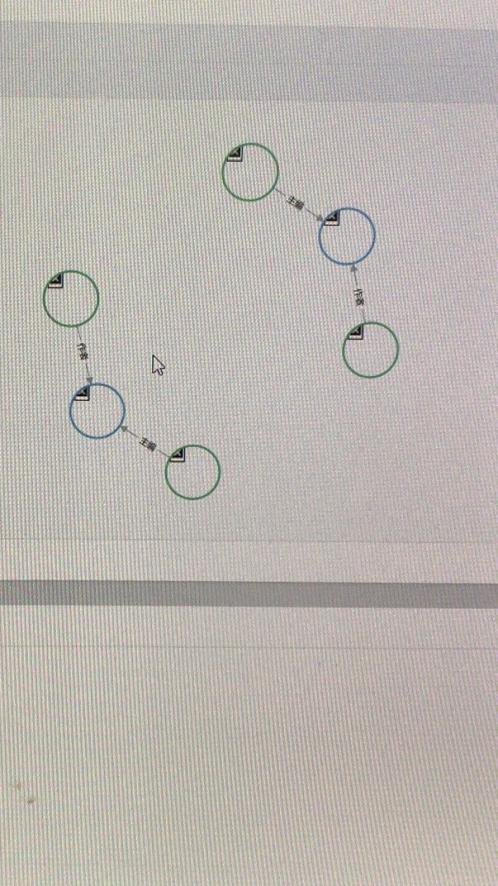
(苏博士)-[:毕业于]->(国防科大),

(张老师)-[:主编]->(权威指南),

(苏博士)-[:作者]->(权威指南)

Match m=(:著作 {名称:"Neo4j权威指南"})--() RETURN m





问题：图片无法正常显示

解决办法：指定节点的相对路径属性存储在服务器中，在本地中读取无法获取。因此创建带图节点有一定难度。

1. **创建图数据：**

CREATE

(张老师:写作成员{名称:"张老师",生日:1970}),

(苏博士:写作成员{名称:"苏博士",生日:1969}),

(国防科大:大学{名称:"国防科大",建校:1876}),

(权威指南:著作{名称:"Neo4j权威指南",出版日期:2014}),

(张老师)-[:毕业于]->(国防科大),

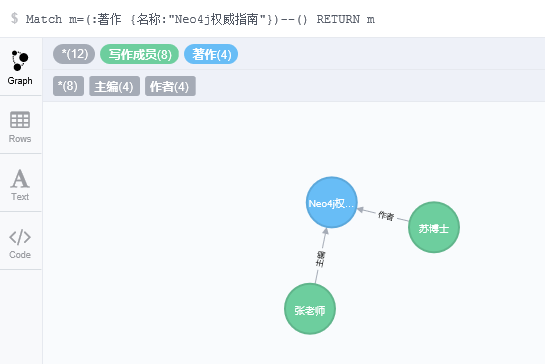
(苏博士)-[:毕业于]->(国防科大),

(张老师)-[:主编]->(权威指南),

(苏博士)-[:作者]->(权威指南)

Match m=(:著作 {名称:"Neo4j权威指南"})--() RETURN m

**查看结果查询1度关系：**



* 创建电影节点：

CREATE(TheMatrix:Movie {title:'The Matrix',released:1999,tagline:'welcome to the Real World'})

CREATE(TheDevilsAdvocate:Movie {title:'The Devil’s Advocate',released:1991,tagline:'welcome '})

CREATE(AFewGoodMen:Movie {title:'A Few Good Men',released:1992,tagline:'to the Real World'})

CREATE(AsGoodAsItGets:Movie {title:'As Good As It Gets',released:1992,tagline:' Real World'})

CREATE(cloudAtlas:Movie {title:'Cloud Atlas',released:1999,tagline:'to the Real '})

* 创建人物节点：与演员导演关系

CREATE

(Keanu:Person {name:'Keanu Reeves',born:1964}),

(Carrie:Person {name:'Carrie',born:1964}),

(Laurence:Person {name:'Laurence',born:1965}),

(Hugo:Person {name:'Hugo',born:1962}),

(LillyW:Person {name:'LillyW',born:1963}),

(LanaW:Person {name:'LanaW',born:1968}),

(JoelS:Person {name:'JoelS',born:1972}),

(Keanu)-[:ACTED\_IN {roles:['Neo']}]->(TheMatrix),

(Keanu)-[:ACTED\_IN {roles:['Alice']}]->(cloudAtlas),

(Keanu)-[:ACTED\_IN {roles:['Leis']}]->(AsGoodAsItGets),

(Carrie)-[:ACTED\_IN {roles:['Trinity']}]->(TheMatrix),

(Laurence)-[:ACTED\_IN {roles:['Morpheus']}]->(TheMatrix),

(Hugo)-[:ACTED\_IN {roles:['Agent Smith']}]->(TheMatrix),

(LillyW)-[:DIRECTED]->(TheMatrix),

(LanaW)-[:DIRECTED]->(TheMatrix),

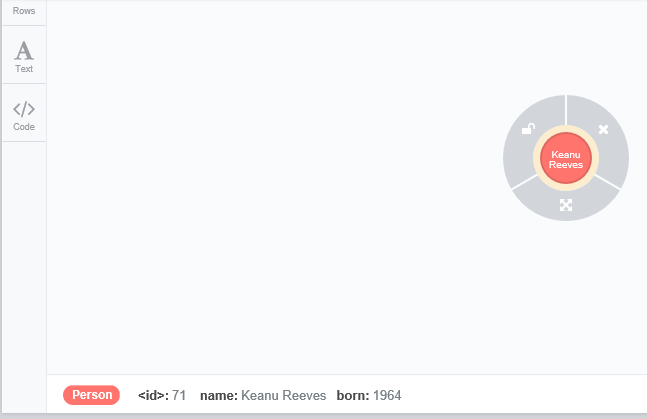
(JoelS)-[:PRODUCED]->(TheMatrix)

1. **检索节点**

* 查找人员

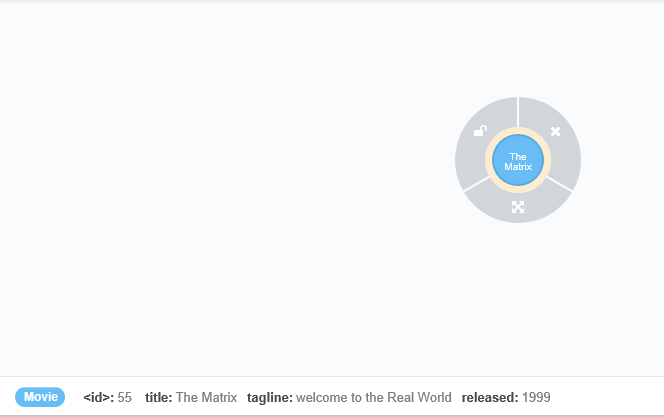
MATCH (Keanu {name: "Keanu Reeves"}) RETURN Keanu

可查看节点属性：



* 查找电影节点

$MATCH (TheMatrix {title: 'The Matrix'}) RETURN TheMatrix



* 随机查找多个人物人名

MATCH(people:Person) RETURN people.name LIMIT 7



* **查找多个电影**

MATCH(nineties:Movie)

WHERE nineties.released > 1990 AND nineties.released < 2000

RETURN nineties.title

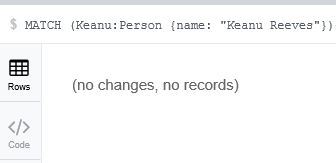


1. **查询关系**

* **查找**Keanu Reeves参演过的电影

MATCH (Keanu:Person {name:”Keanu Reeves”})-[:ACTED\_IN]->(KeanuReevesMovies) RETURN Keanu,KeanuReevesMovies

MATCH (Hugo:Person {name:'Hugo'})-[:ACTED\_IN]->(HugoMovies) RETURN Hugo,HugoMovies



* 查找谁导演了电影THE Matrix

MATCH (TheMatrix {title: ‘The Matrix’})<-[:DIRECTED]-(directors)

RETURN directors.name

* 查找与Keanu Reeves同出演过电影的人

MATCH (Keanu:Person {name:”Keanu Reeves”})-[:ACTED\_IN]->(m)<-[:ACTED\_IN]-(coActors)

RETURN coActors.name

* 查找与电影The Matrix相关的所有人

MATCH (people:Person)-[relatedTo]-(:Movie {title: “The Matrix”})

RETURN people.name, Type(relatedTo), relatedTo

* 查询关系路径

1. 查找与演员Keanu Reeves存在4条及以内关系的任何演员和电影

MATCH (Keanu:Person {name:”Keanu Reeves”})-[\*1..4]-(hollywood)

RETURN DISTINCT hollywood

1. 查找与演员Keanu Reeves与导演LillyW最短关系路径

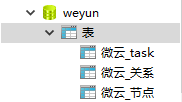
MATCH p=shortestPath(

(Keanu:Person {name:”Keanu Reeves”})-[\*]-(LillyW:Person {name:”LillyW”}))

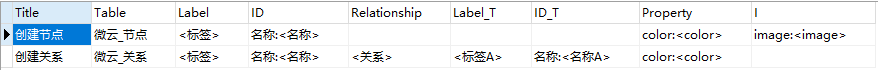
RETURN p

1. **使用导入精灵导入mysql**

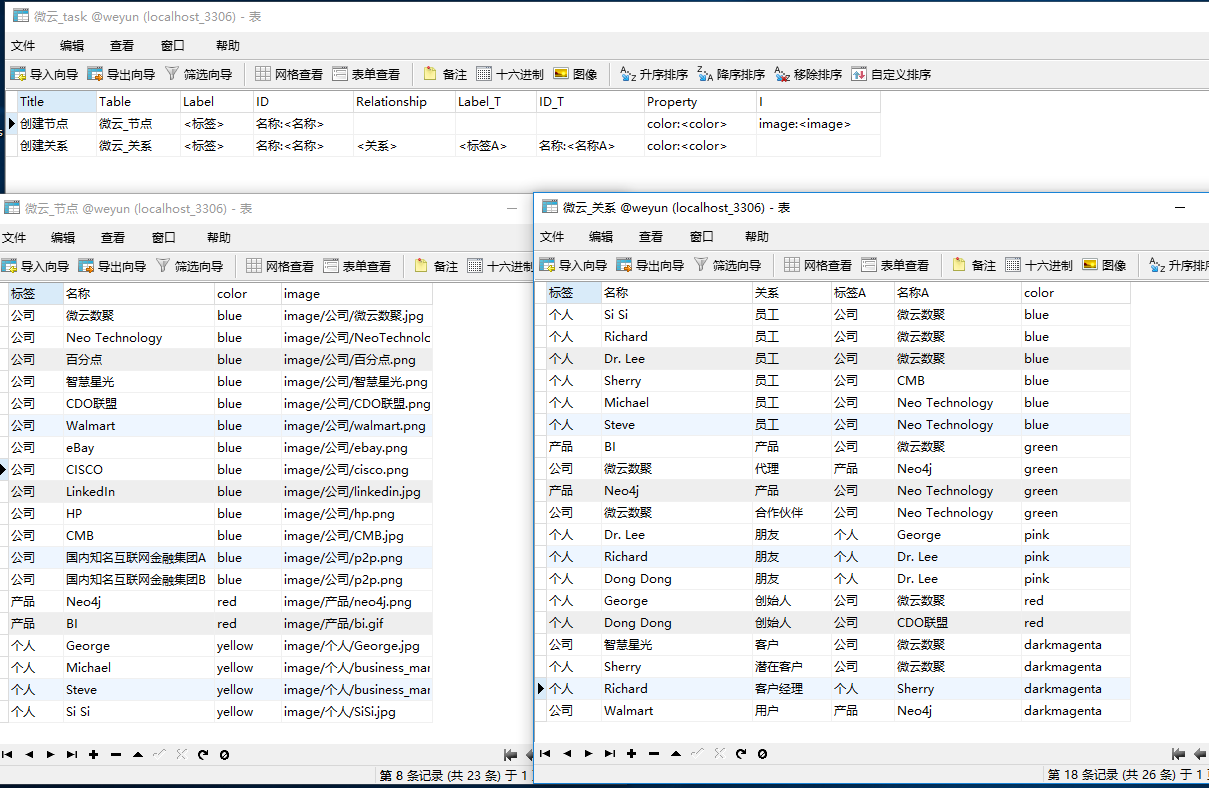
需要在http://we-yun.com/neo4jguide/微云.sql下载weyun.sql，该数据库包含如下表:



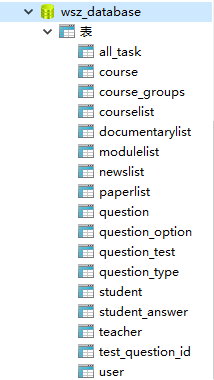
微云\_task是作为使用导入精灵导入数据库的模板，要按照如下表创建节点和创建关系：



weyun数据库中三个表结构如下图：



微思政的数据库包含下表：

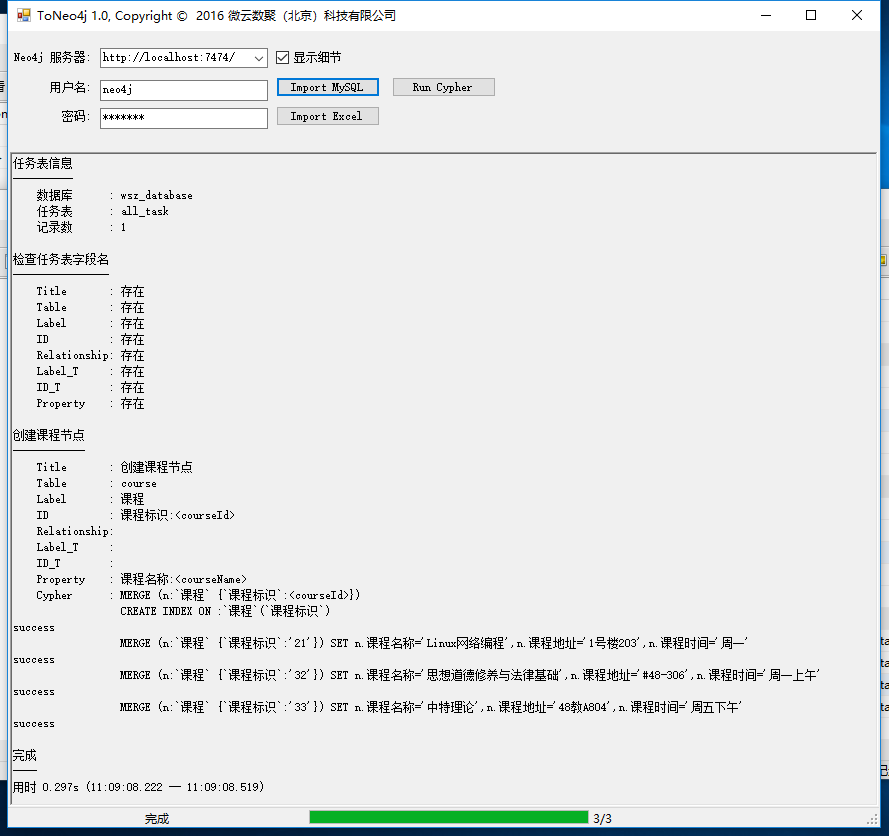


其中**all\_task表**是按照**[微云\_task]**格式的**\_task**后缀创建的，表结构如下：

导入课程节点：



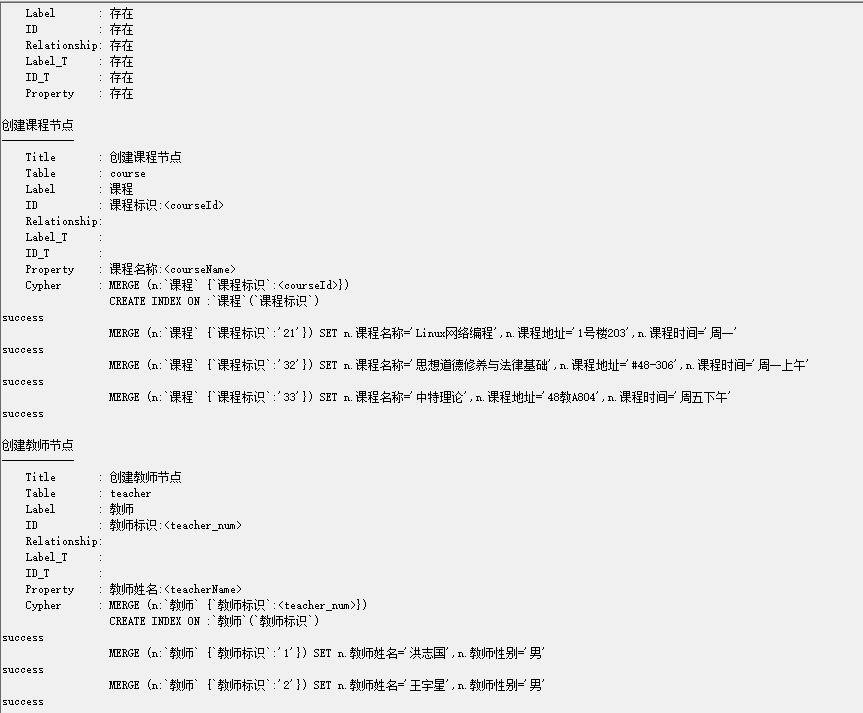
使用导入精灵导入课程节点：



导入教师节点：

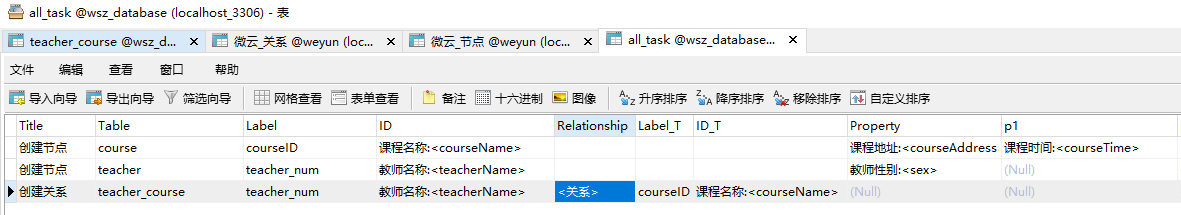


使用导入精灵导入教师节点：

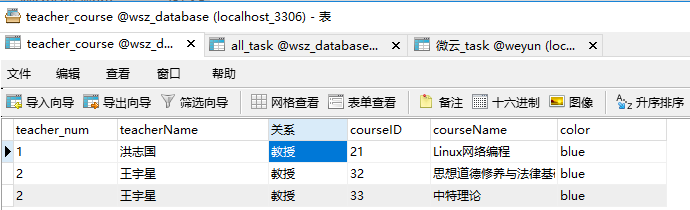


创建teacher\_course关系:

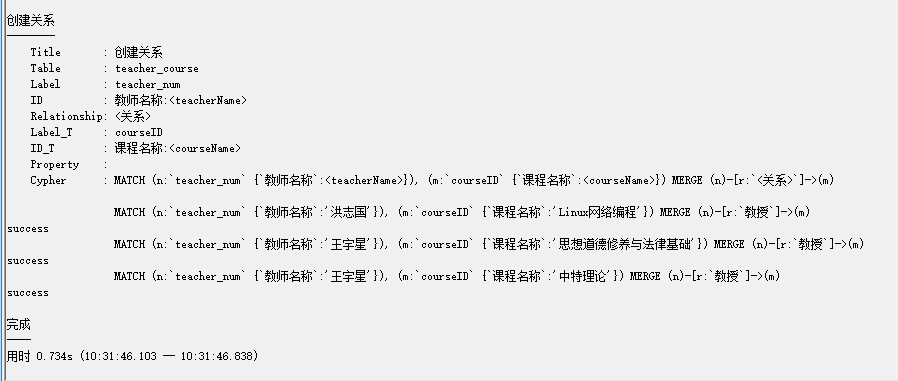
all\_task表如下



关系表teacher\_course设计如下：



使用导入精灵导入关系：



可视化界面如下：



