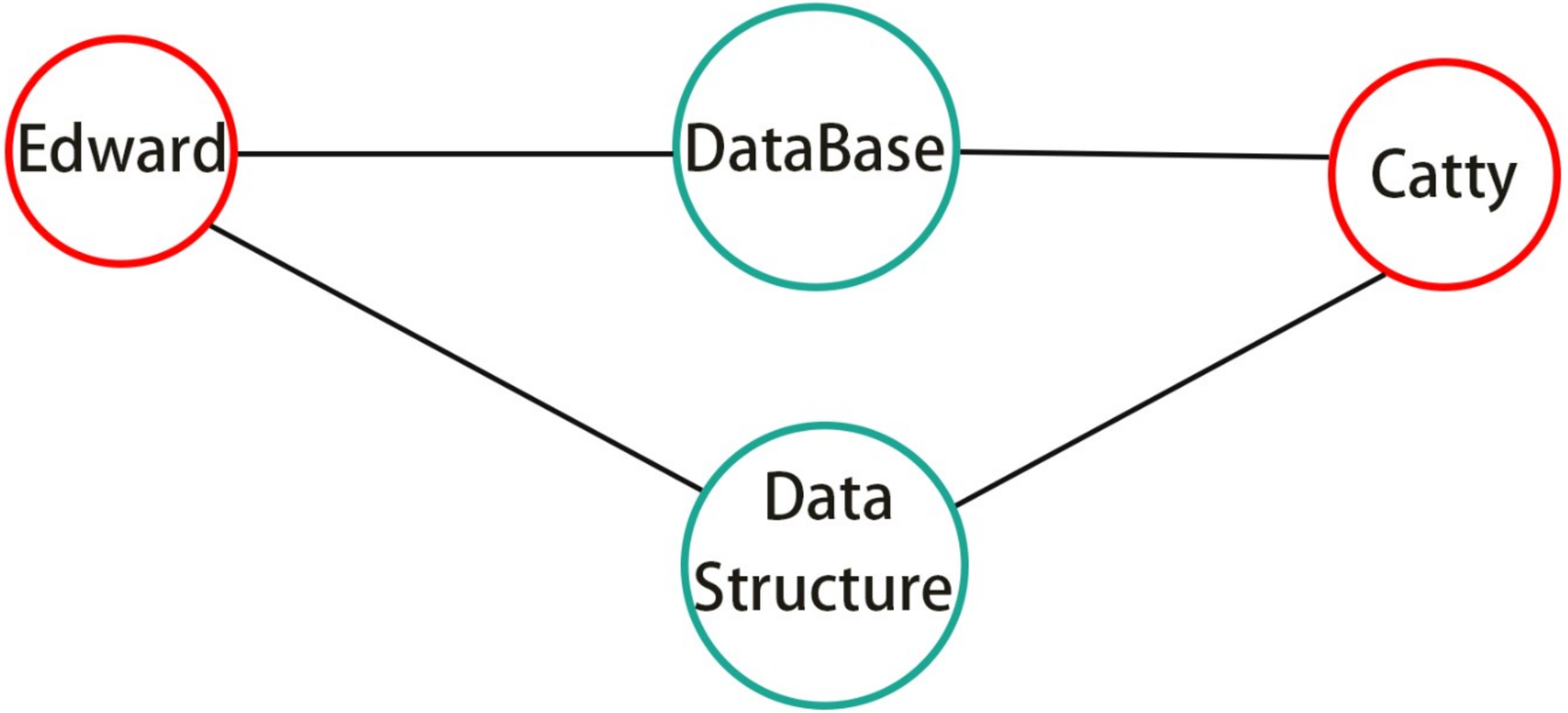
霍格沃兹测试学院 - 测试开发工程师的黄埔军校

Neo4j 数据库

MrDong

目录

Neo4j




田馥甄
张杰
萧敬腾
金莎
林怡凤
关喆
张怀秋
郭静
李玖哲
林志颖
陈乃冷
方文山
蔡健雅
张卓楠
陈信宏
蔡卓妍
汪苏泷
胡彦斌
林俊杰
暧昧
好友
好友
师兄
朋友
朋友
老乡
好友
朋友
朋友
撞脸
朋友
好友
前绯闻朋友

Neo4j 定义

- Neo4j 是一个高性能的 ,NOSQL 图形数据库，它将结构化数据存储在网络上而不是表中

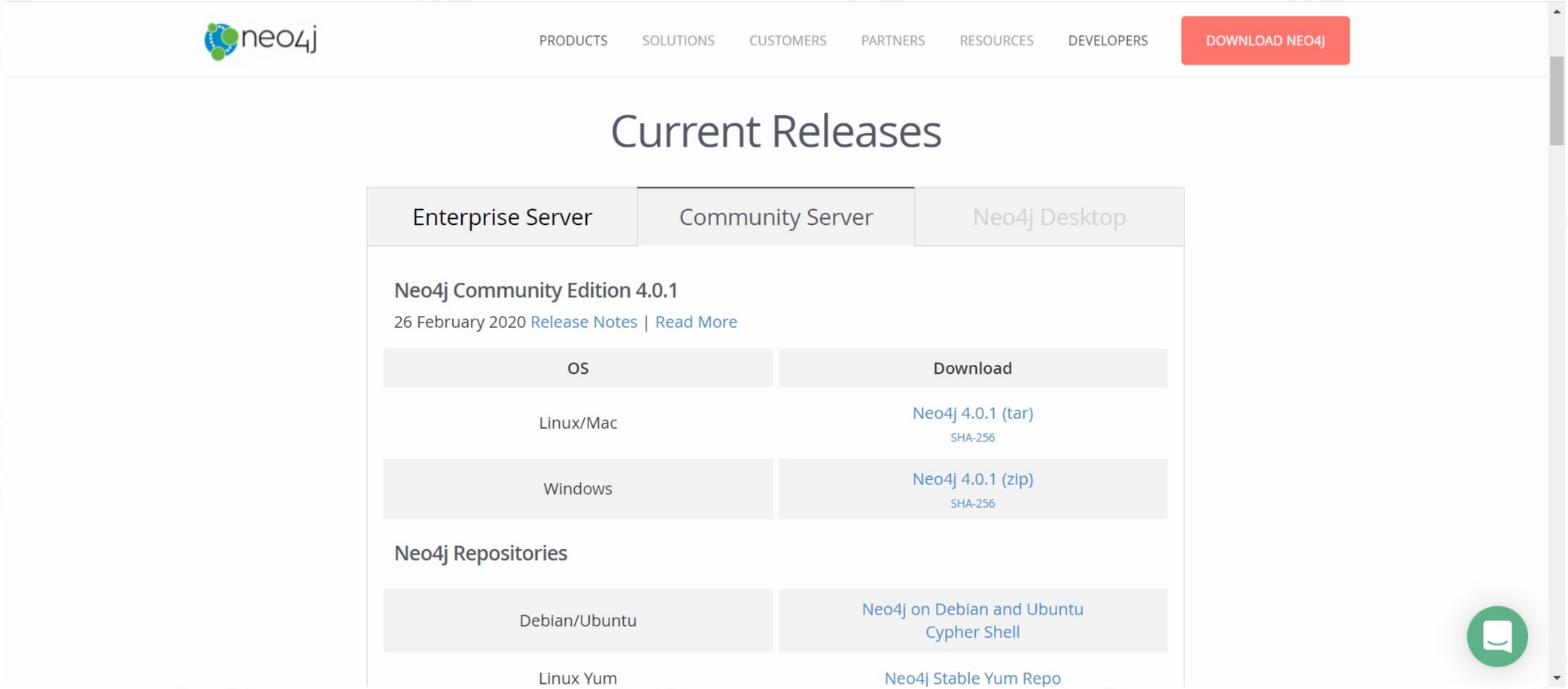
Neo4j 特点

- 非关系型数据库
- NoSql
- CQL
- 遵循 SQL 语法，简单，且人性化

安装与使用

下载

- https://neo4j.com/download-center/#community

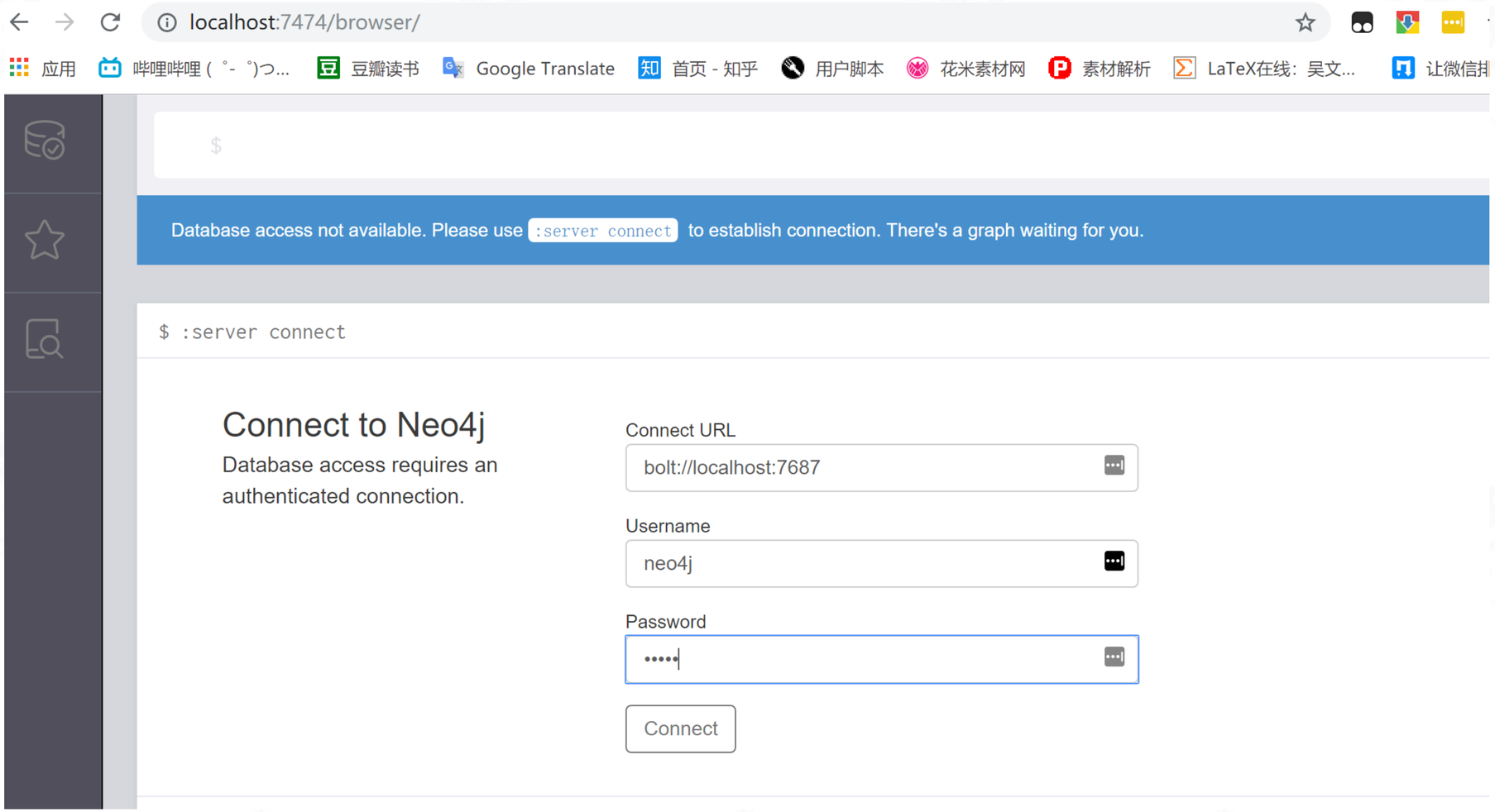


下载与使用

- 运行 Neo4j
  - neo4j console
- 输入默认账号密码
  - neo4j
- 修改账号密码

```
Microsoft Windows [版本 10.0.18363.657]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\yuruo>neo4j console
2020-03-03 23:19:40.854+0000 INFO  ======== Neo4j 4.0.1 ========
2020-03-03 23:19:40.858+0000 INFO  Starting...
2020-03-03 23:19:44.675+0000 INFO  Bolt enabled on localhost:7687.
2020-03-03 23:19:44.676+0000 INFO  Started.
2020-03-03 23:19:45.307+0000 INFO  Remote interface available at http://localhost:7474/
```



```
$ :server connect
```

Connect to Neo4j

Database access requires an authenticated connection.

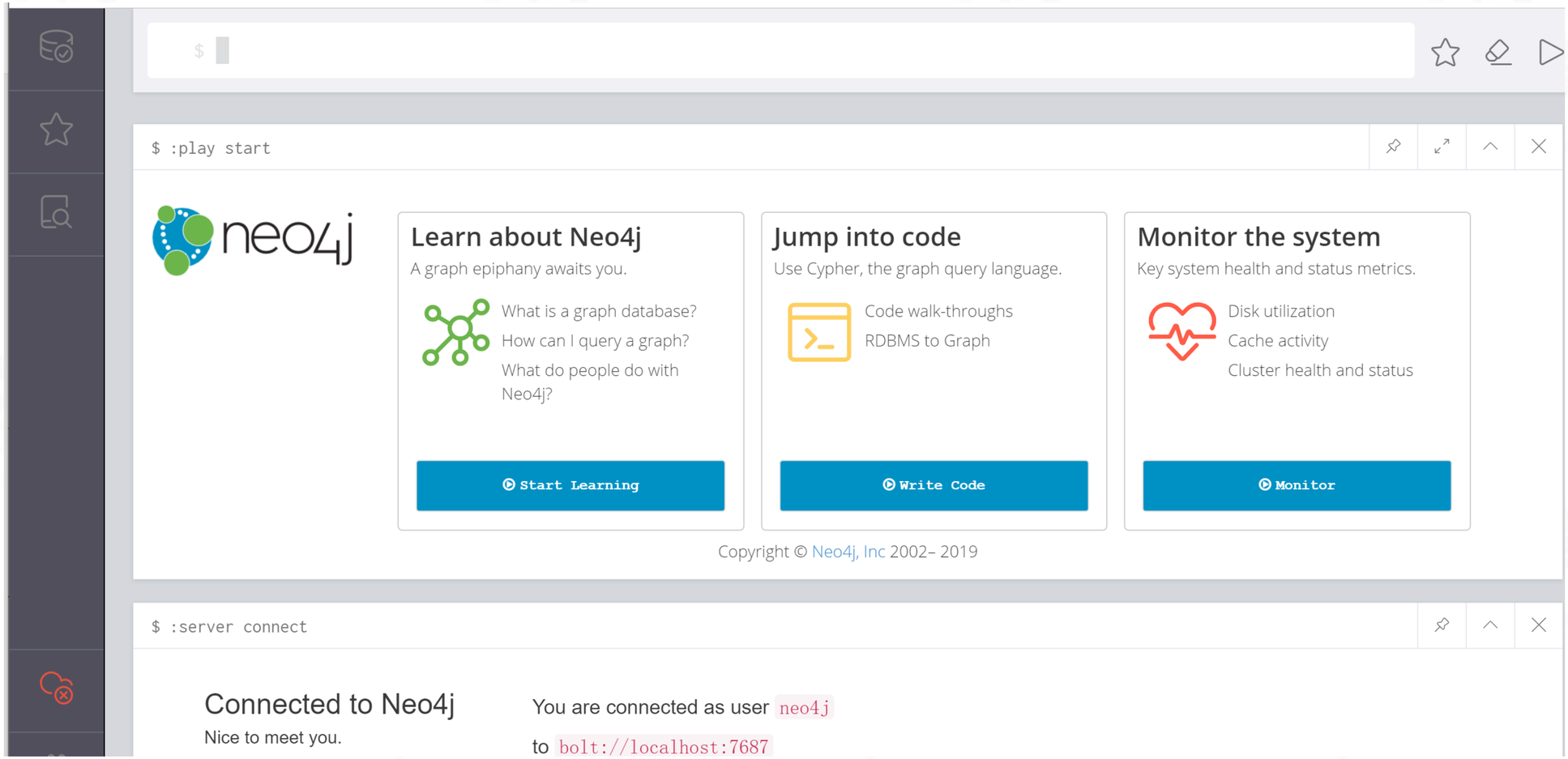
New password

••••••••

Repeat new password

••••••••

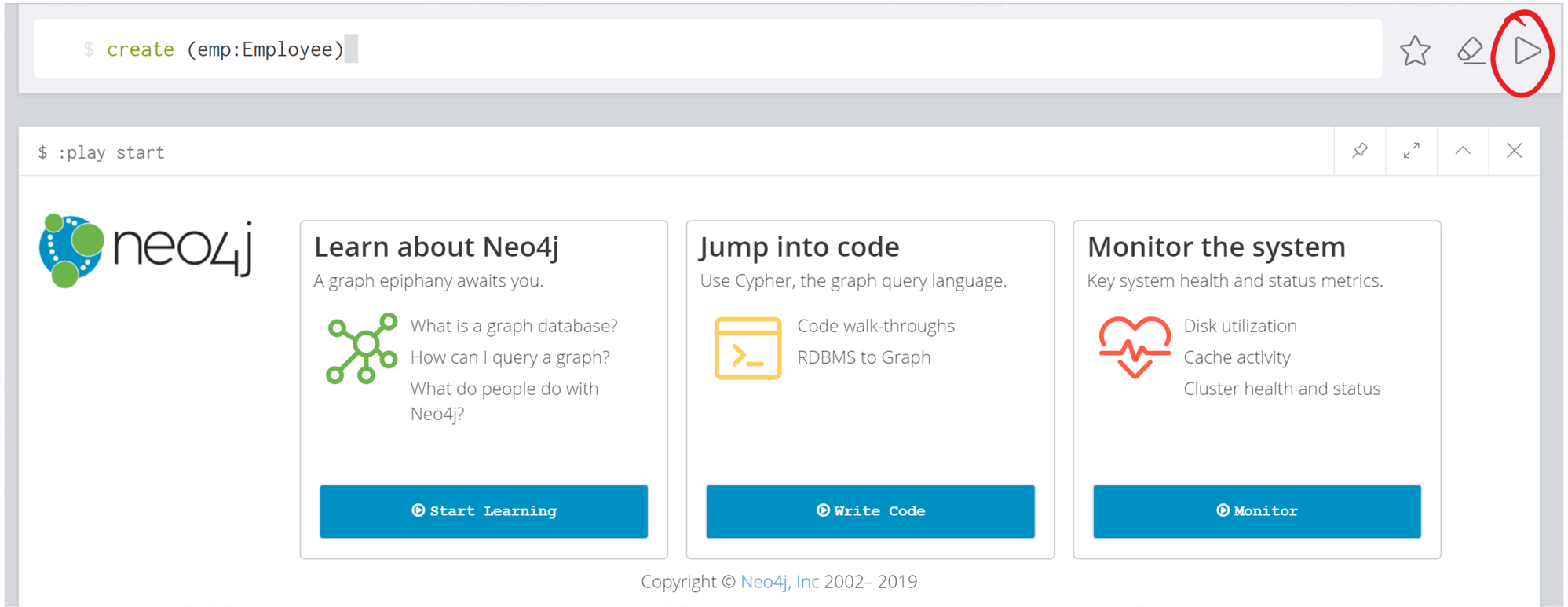
Change password



创建节点

创建标签节点

- 打开浏览器
- 创建 emp 节点，标签为 Employee
  - create(emp:Employee)
- 创建 XiaoHong 节点，带有属性
  - CREATE(XiaoHong:People{ no:1,age:10,gender:”M”})



```
$ CREATE (XiaoHong:People { no:1,age:10,gender:"M" })
```

$ CREATE (XiaoHong:People { no:1,age:10,gender:"M" })

Table

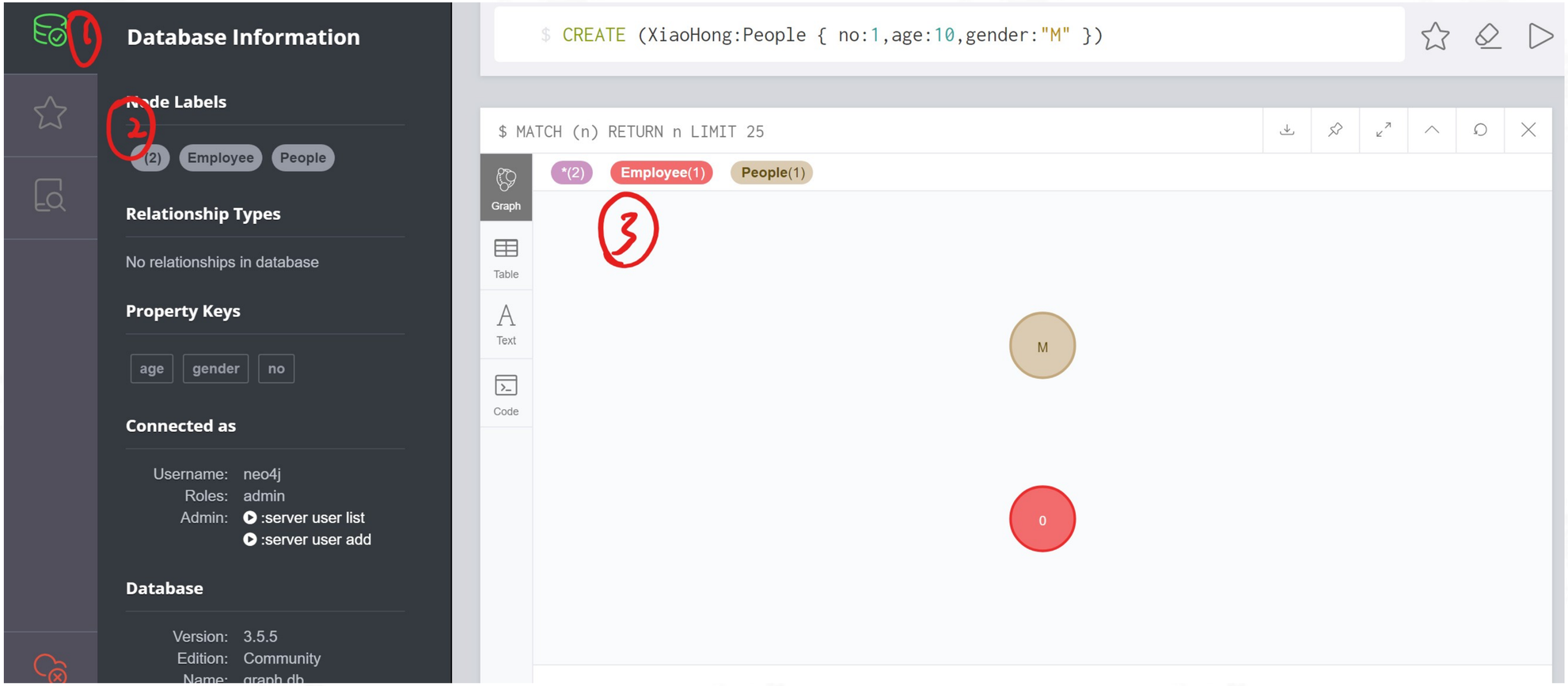
Code

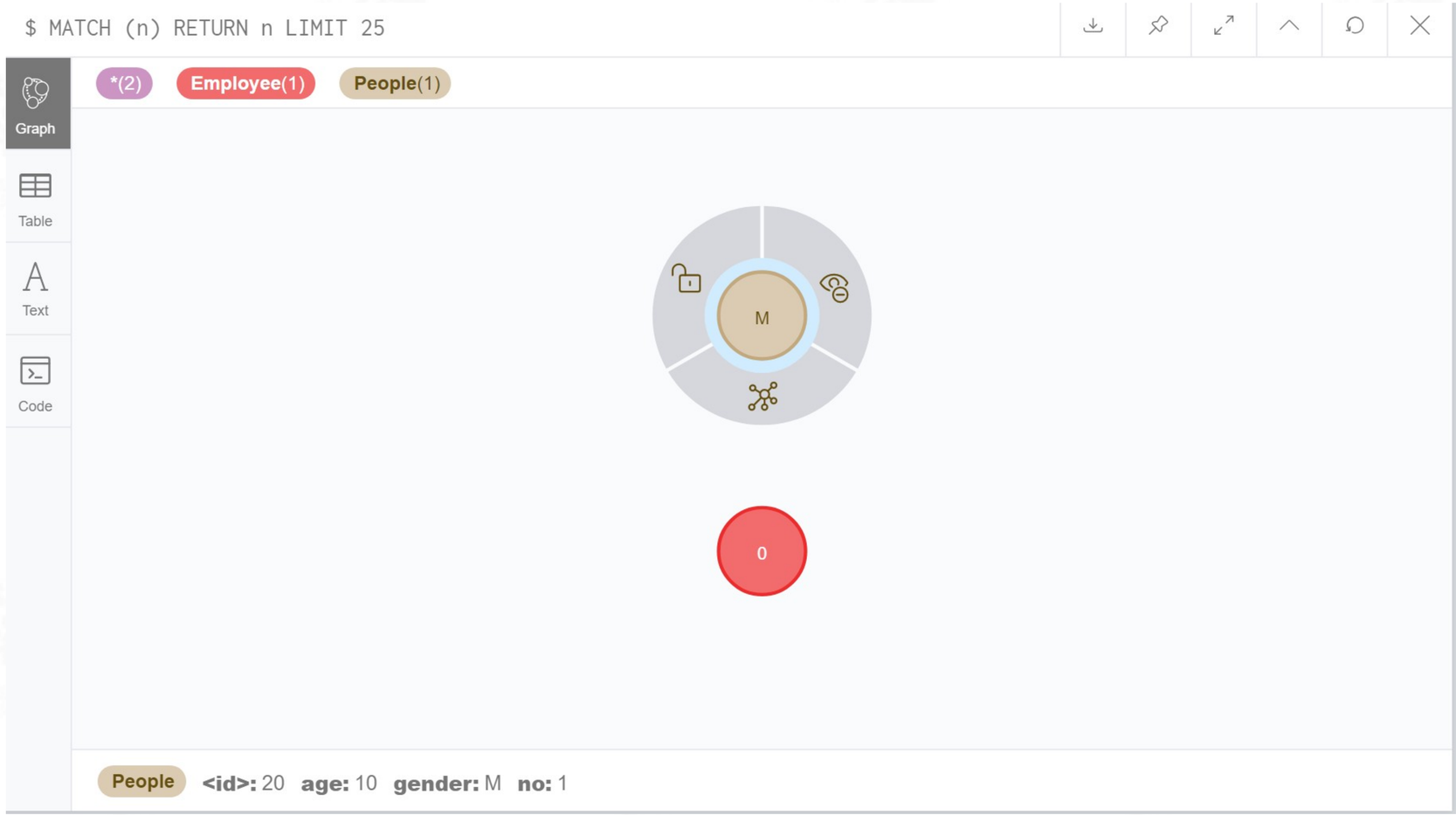
Added 1 label, created 1 node, set 3 properties, completed after 94 ms.

Added 1 label, created 1 node, set 3 properties, completed after 94 ms.

什么是节点


田馥甄
张杰
萧敬腾
金莎
林怡凤
关喆
张怀秋
暧昧
好友
好友
师兄
朋友
老乡
蔡健雅
朋友
好友
郭静
林俊杰
朋友
朋友
张卓楠
好友
前绯闻朋友
朋友
撞脸
朋友
陈信宏
李玖哲
林志颖
陈竹冷
方文山
蔡卓妍
汪苏泷
胡彦斌





添加属性

创建带属性的节点

- CREATE (book:Book {title:“book1",pages:340,price:250})

添加属性

- match (book{title:”book1 ”})
- set book.name = “hello”
- return book

```
$ create(book:Book{title:"book1",pages:340,price:250})
```

```
$ create(book:Book{title:"book1",pages:340,price:250})
```

Table

Code

Added 1 label, created 1 node, set 3 properties, completed after 149 ms.

Added 1 label, created 1 node, set 3 properties, completed after 149 ms.

```
1 match (book{title:"book1"})
2 set book.name="hello"
3 return book
```

$ match (book{title:"book1"}) set book.name="hello" return book

Graph
Table
Text
Code

*(1) Book(1)

Book <id>: 0 name: hello pages: 340 title: book1

查询

检索节点的 age 属性

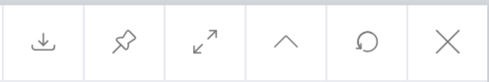
- MATCH 需要与 RETURN 一起使用
  - MATCH (XiaoHong:People)
  - RETURN XiaoHong.age

检索节点的所有属性

- MATCH 与 RETURN 不能单独使用
  - MATCH (XiaoHong:People)
  - RETURN XiaoHong

```
1 MATCH (XiaoHong:People)
2 RETURN XiaoHong.age
```

$ MATCH (XiaoHong:People) RETURN XiaoHong.age



Table

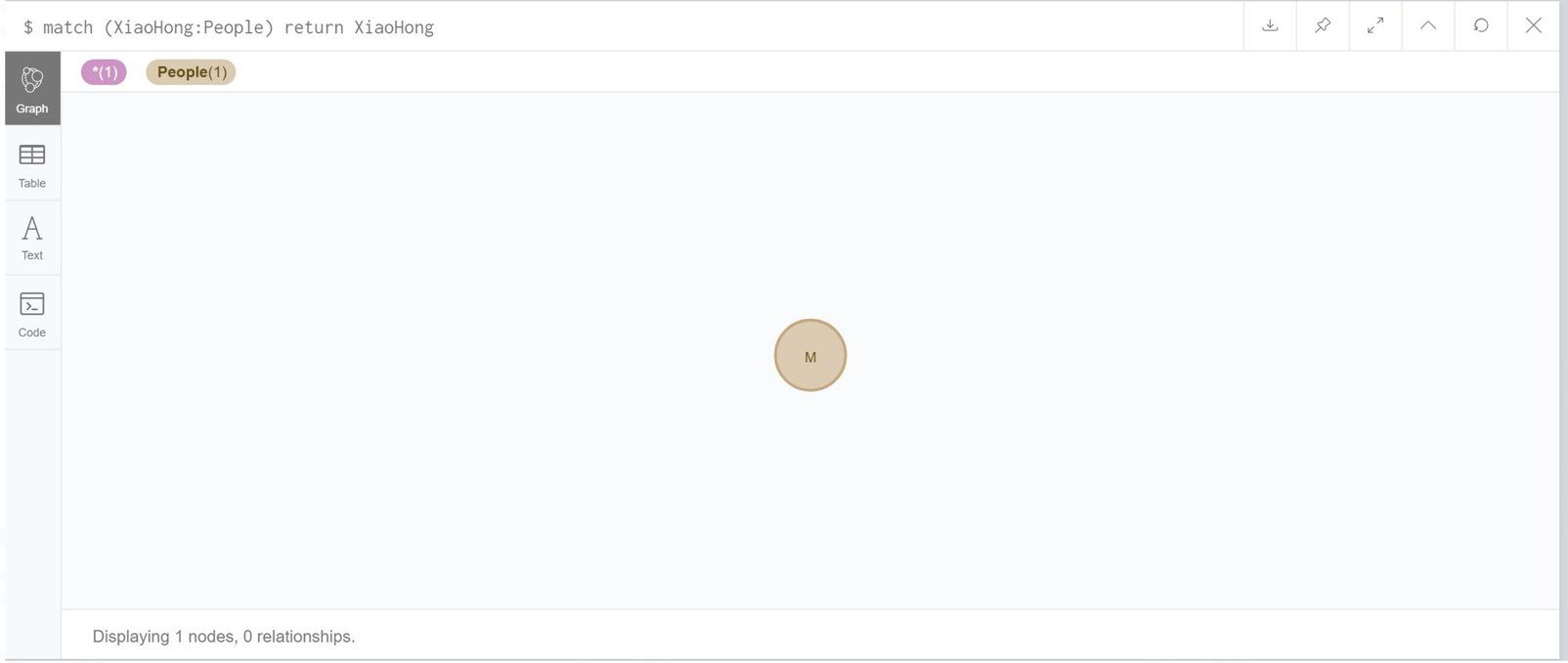
Text

Code

XiaoHong.age
10

Started streaming 1 records after 1 ms and completed after 37 ms.

```
1 match (XiaoHong:People)
2 return XiaoHong
```



关系

什么是关系


田馥甄
张杰
萧敬腾
金莎
林怡凤
关喆
张怀秋
暧昧
好友
好友
师兄
朋友
老乡
蔡健雅
郭静
朋友
好友
张卓楠
林俊杰
朋友
朋友
李玖哲
好友
朋友
撞脸
朋友
陈信宏
林志颖
陈竹冷
方文山
蔡卓妍
汪苏泷
胡彦斌

创建关系

- CREATE (p1:Profile1)-[r1:LIKES]->(p2:Profile2)

```
$ CREATE (p1:Profile1{name:'XiaoHong'})-[r1:Friends]->
  (p2:Profile2{name:'LiMei'})
```

$ CREATE (p1:Profile1{name:'XiaoHong'})-[r1:Friends]->(p2:Profile2{name:'LiMei'})

Table

Code

Added 2 labels, created 2 nodes, set 2 properties, created 1 relationship, completed after 11 ms.

Added 2 labels, created 2 nodes, set 2 properties, created 1 relationship, completed after 11 ms.

```
1 match(a:People),(b:People)
2 where a.name="xiaohong" And b.name="WangWu"
3 create (a)-[r1:Friends]->(b)
```

$ match(a:People),(b:People) where a.name="xiaohong" And b.name="WangWu" create (a)-[r1:…

Table

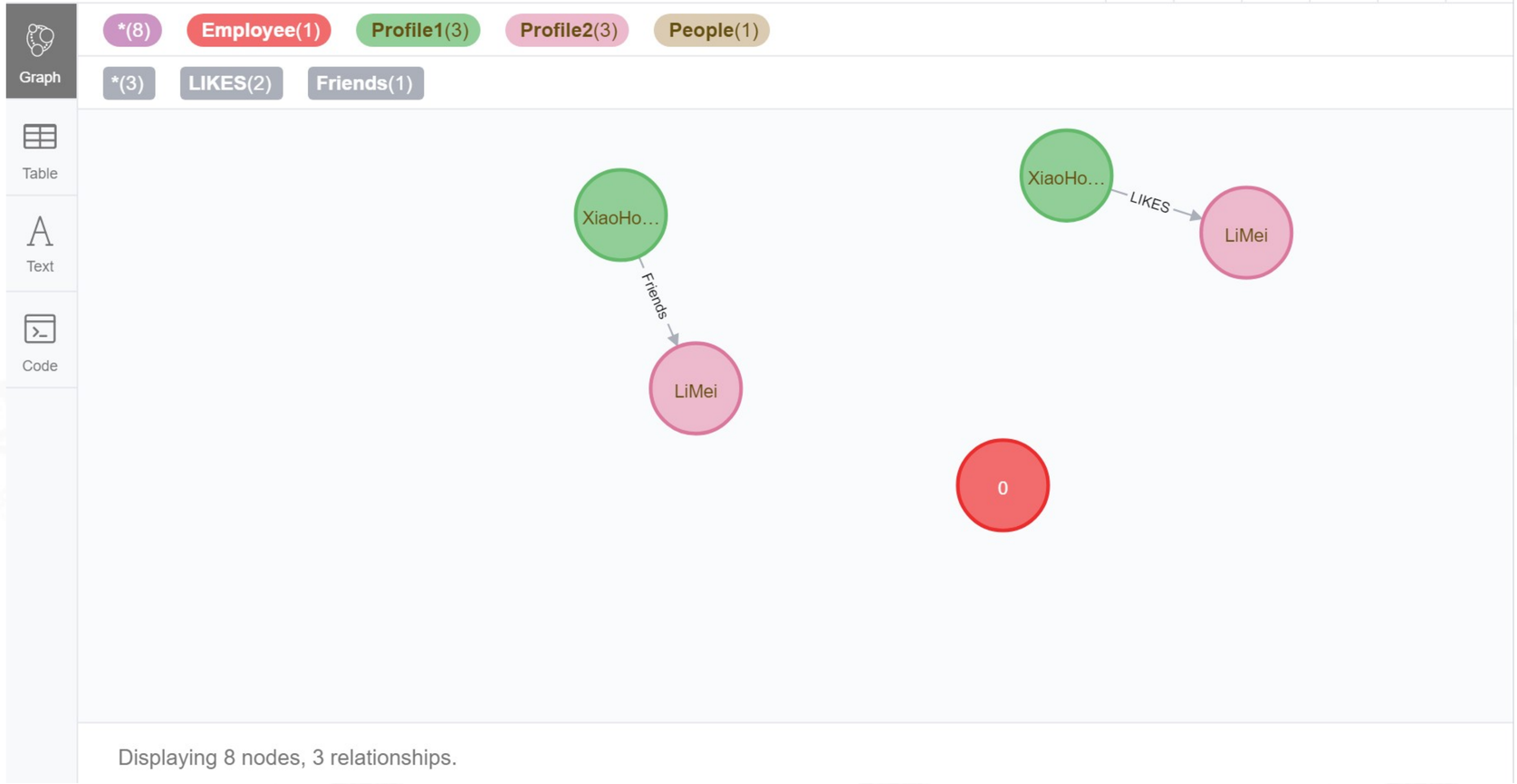
Code

Created 1 relationship, completed after 5 ms.

Created 1 relationship, completed after 5 ms.

```
$ CREATE (p1:Profile1{name:'XiaoHong'})-[r1:Friends]->
  (p2:Profile2{name:'LiMei'})
```

```
$ MATCH (n) RETURN n LIMIT 25
```



删除节点和关系

删除相应属性下的节点

- MATCH (e: People) DELETE e

删除带有关系的节点

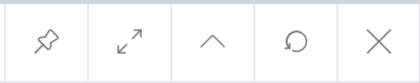
- MATCH (cc: Profile1)-[Friends]->(c:Profile2)
- DELETE cc,c,Friends

删除相应属性下的节点

- MATCH (e: People) DELETE e

```
$ MATCH (e: People) DELETE e
```

$ MATCH (e: People) DELETE e



Table

Code

Deleted 1 node, completed after 1 ms.

Deleted 1 node, completed after 1 ms.

```
1 MATCH (cc: Profile1)-[Friends]-(c:Profile2)
2 DELETE cc,c,Friends
```

$ MATCH (cc: Profile1)-[Friends]-(c:Profile2) DELETE cc,c,Friends

Table

Code

Deleted 6 nodes, deleted 3 relationships, completed after 12 ms.

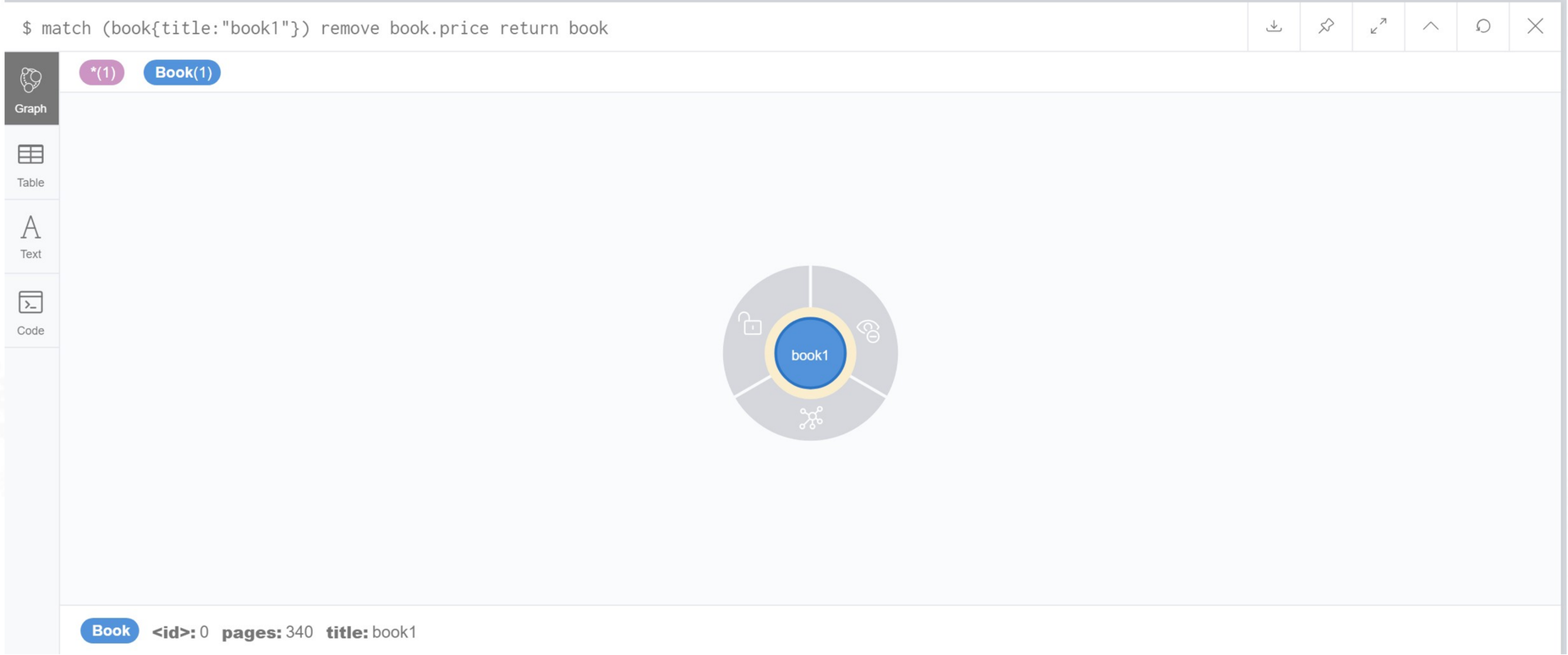
Deleted 6 nodes, deleted 3 relationships, completed after 12 ms.

删除属性

删除属性

- match (book{title:”book1 ”})
- Remove book.price
- Return book

```
1 match (book{title:"book1"})
2 remove book.price
3 return book
```



Delete 与 Remove

- DELETE 操作用于删除节点和关联关系
- REMOVE 操作用于删除标签和属性

额外补充 1

- 删除标签：
  - MATCH (m:Movie)
  - REMOVE m:Picture
- 排序：
  - MATCH (emp:Employee)
  - RETURN emp.empid,emp.name,emp.salary,emp.deptno
  - ORDER BY emp.name DESC

额外补充 2

- 更高级的查询：
  - MATCH (emp:Employee)
  - WHERE emp.name = 'Abc'
  - RETURN emp