# Windows下NoSql图形数据库-Neo4j

安装+环境变量配置+服务注册+访问

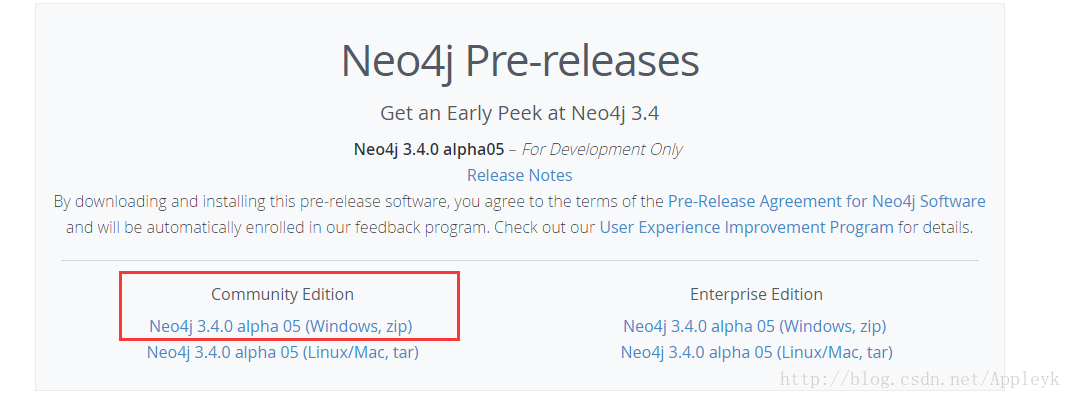
## 安装

### 查看是否安装JDK

依赖Java的JVM虚拟机，因此，保证系统中已安装了JDK



### 下载

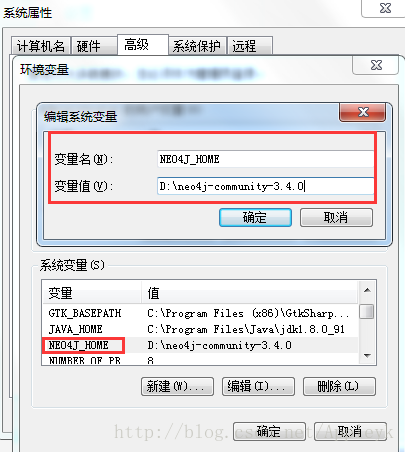
https://neo4j.com/download-center/#releases

### 解压后重命名放置任意盘符下

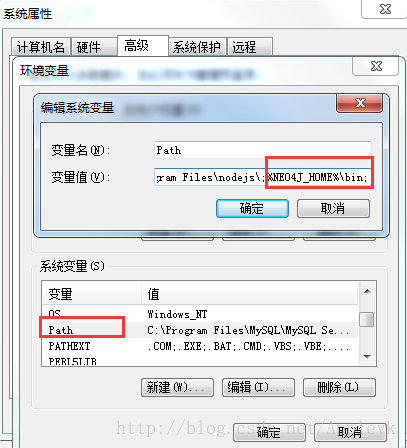
D:\DevelopmentSoftware\Neo4j-community-3.4.6

### 系统环境变量配置

NEO4J\_HOME = D:\neo4j-community-3.4.0



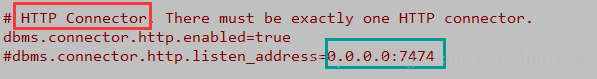
（2）Path = %NEO4J\_HOME%\bin;



### 验证Neo4j的安装和环境变量配置是否成功

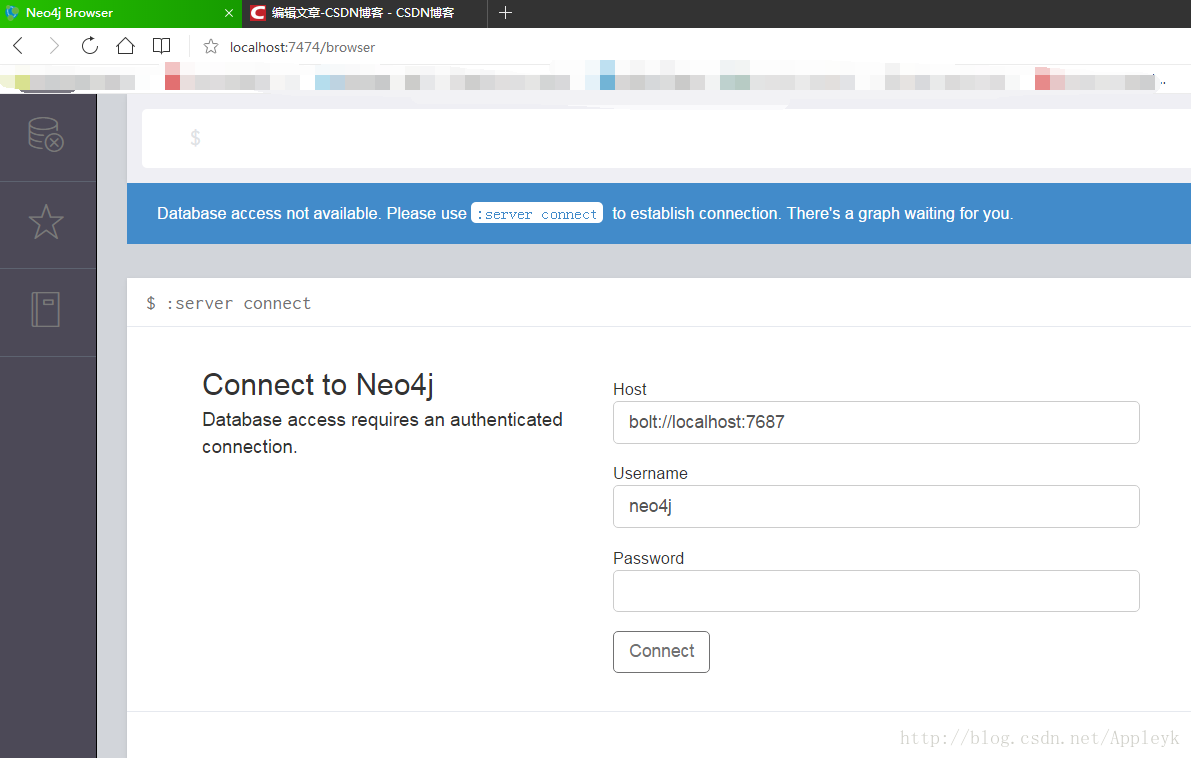
（1）CMD管理员身份运行neo4j.bat console





### 以HTTP连接器的形式（三种连接方式中的一种）访问Neo4j数据库

http://localhost:7474   默认跳转到  http://localhost:7474/browser



**默认用户：neo4j**

**默认密码：neo4j**

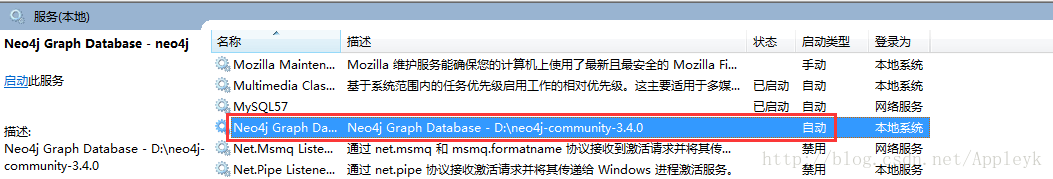
### 注册Neo4j服务

注册服务

neo4j install-service



本地服务查看



### 开启Neo4j服务



### 停止、重启、查询Neo4j服务



### Neo4j图形数据库简单说明

  图形数据库名字的由来其实与其在底层的存储方式有关，Neo4j底层会以图的方式把用户定义的节点以及关系存储起来，通过这种方式，可以高效的实现从某个节点开始，通过节点与节点之间的关系，找出两个节点间的联系。

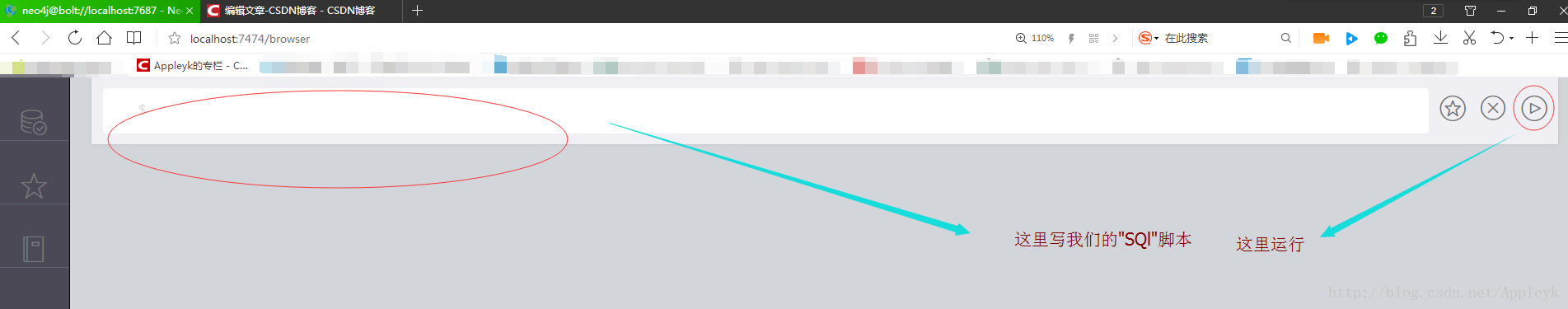
## 基本使用

### 示例一

#### 创建节点

数据场景：节点1（男） +关系（爱）+节点2（女）

打开我们的Neo4J，准备好编写脚本来创建我们的第一个graph data！



##### 创建节点1

姓名 ：鱼暖暖（节点的名称即节点的变量名,下面我们用简写的英文字母表示）

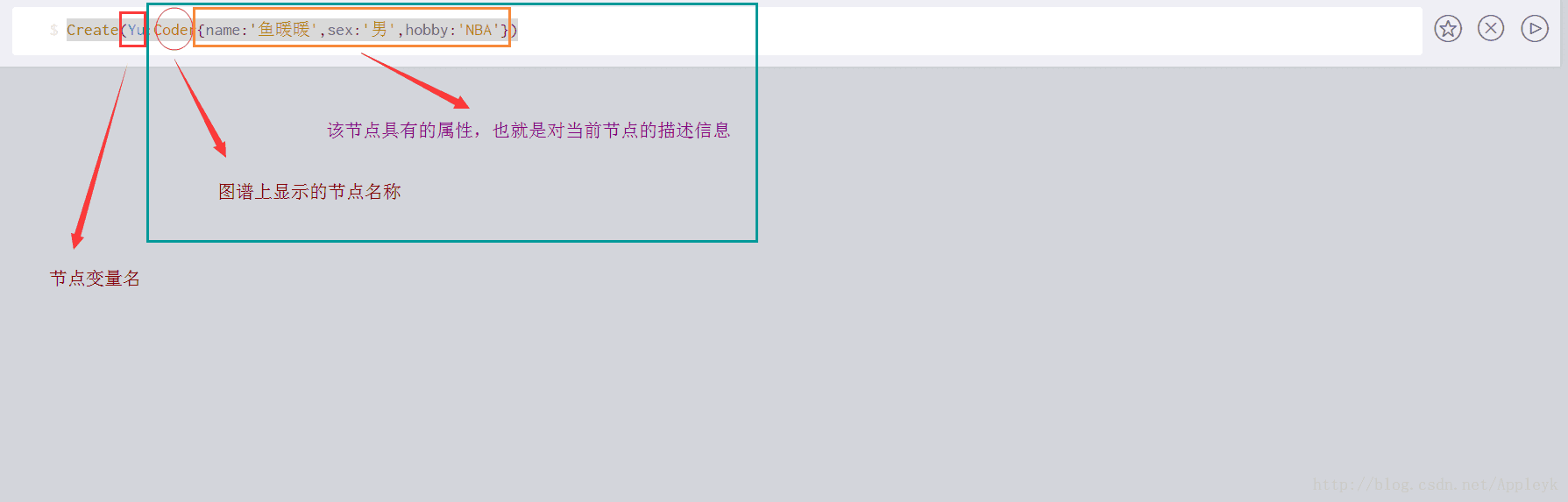
性别 ：男

职业 ：程序员（节点的类别，可作为节点属性的data）

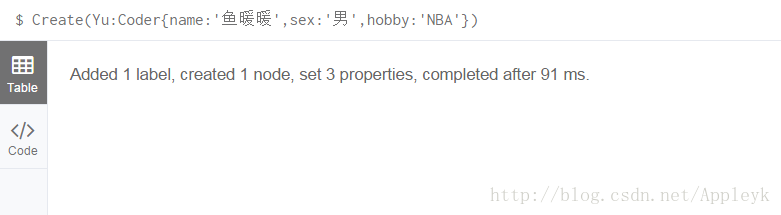
爱好 ：NBA

利用Cypher语句创建我们上面的节点

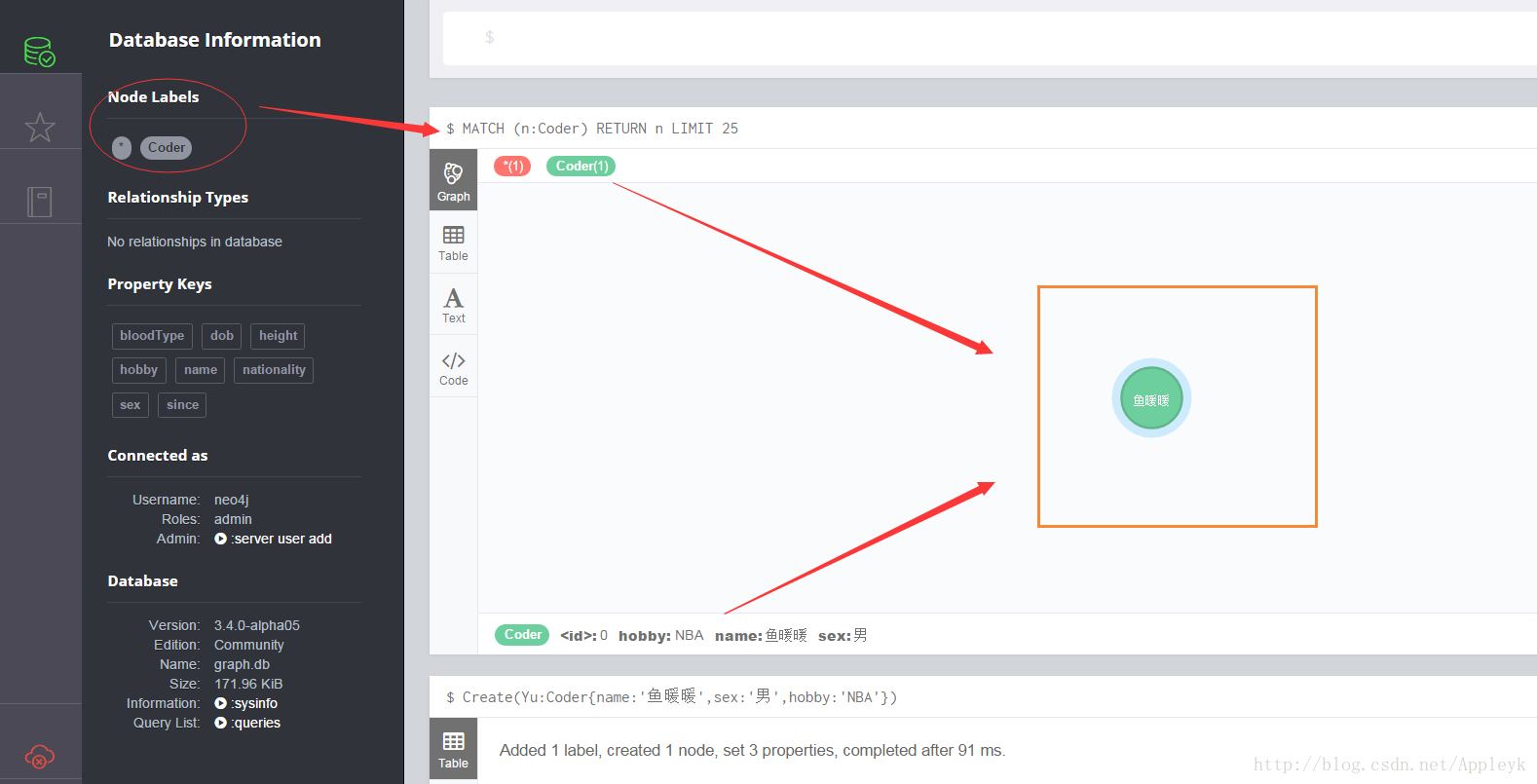
Create(Yu:Coder{name:'鱼暖暖',sex:'男',hobby:'NBA'})



只要你熟悉JOSN，创建一个这样的节点很简单



既然是Neo4j是图形数据库，那么我们就来看一下，这个节点在Neo4j的database里面是如何呈现在我们眼前的



##### 创建节点2

姓名 ：凉暖暖（节点的名称即节点的变量名,下面我们用简写的英文字母表示）

性别 ：女

职业 ：摄影师（节点的类别，可作为节点属性的data）

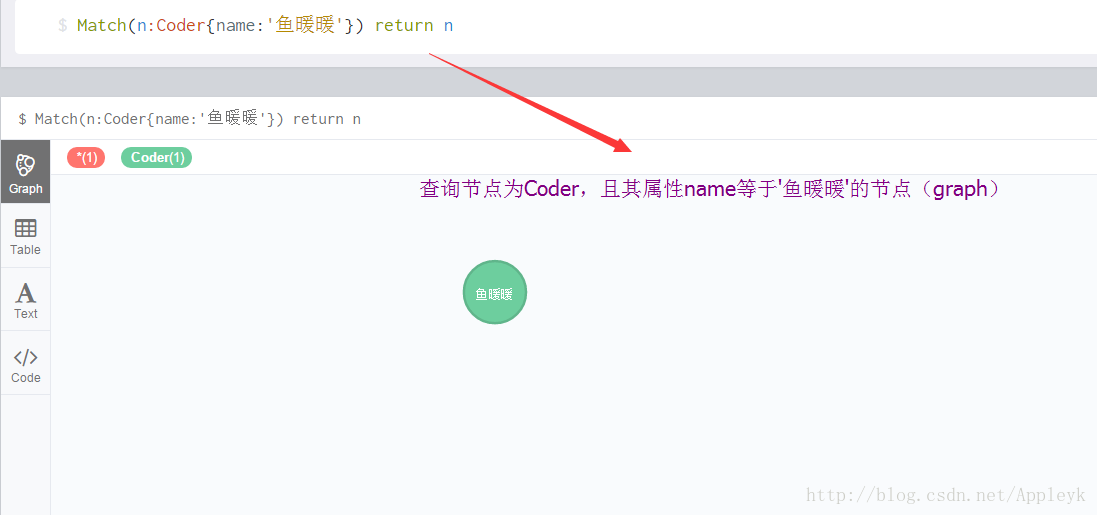
爱好 ：拍照

同上创建方式



##### 查询节点1

Match相当于Sql数据库中的select查询，不过在NoSql中，表示匹配、查找



##### 创建关系（爱）连接节点1和节点2

Match(n:Coder{name:'鱼暖暖'})  
Match(m:Photographer{name:'凉暖暖'})  
Merge(n)-[:LOVE{years:'five years more'}]->(m)

1.查到节点1 变量名：n

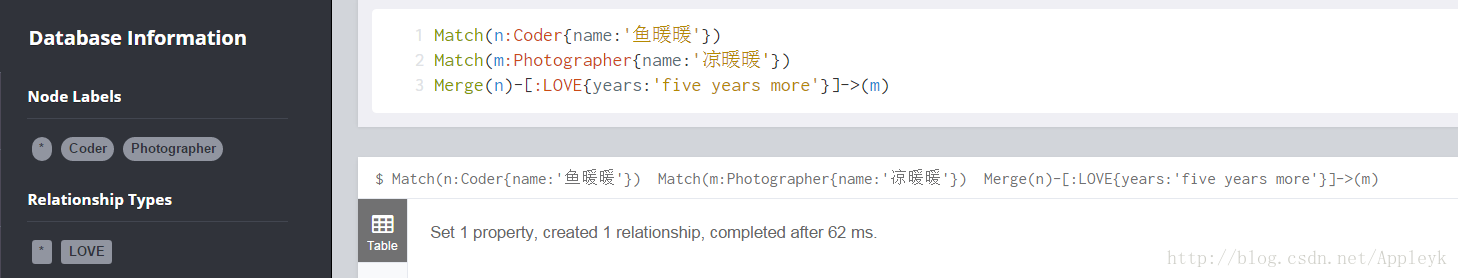
2.查到节点2 变量名：m

3.关系连接

4.防止Match的节点不存在，关系连接的时候用Merge：如果节点不存在创建

5.关系：[:关系data]

6.连接：  （n）-[:关系]->（m）

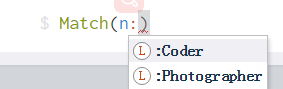


左侧一栏关系类型中，已经出现了"Love",点开查看

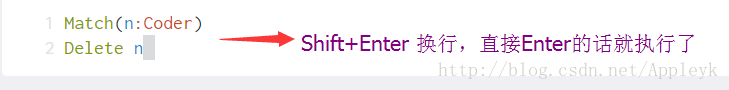


##### 删除节点

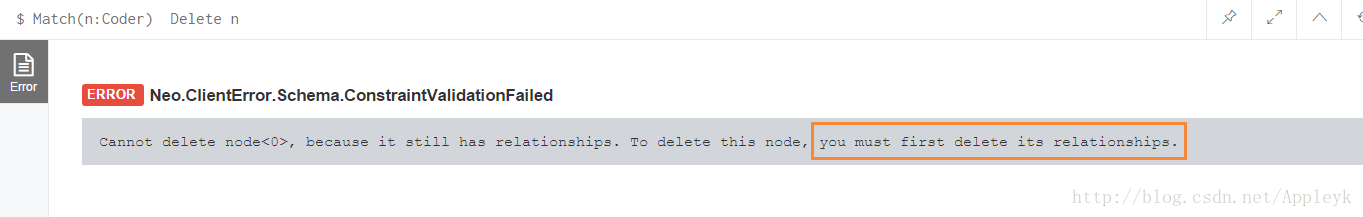
###### (1)智能提示，匹配出所有节点



###### (2)随便选择一个节点，这里选Coder

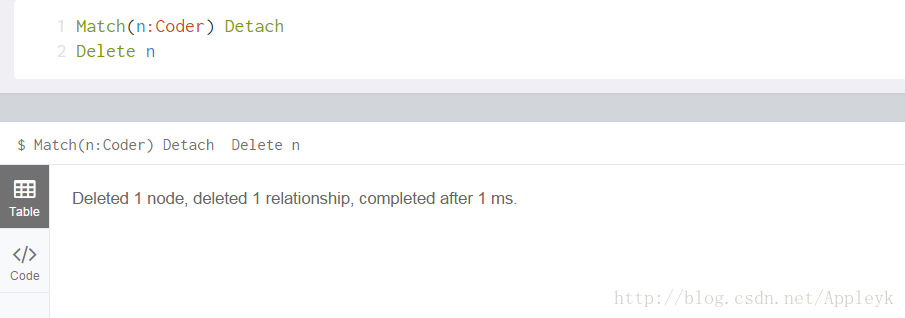


这样是删不掉的，为什么，别忘了，这个节点可是有关系的



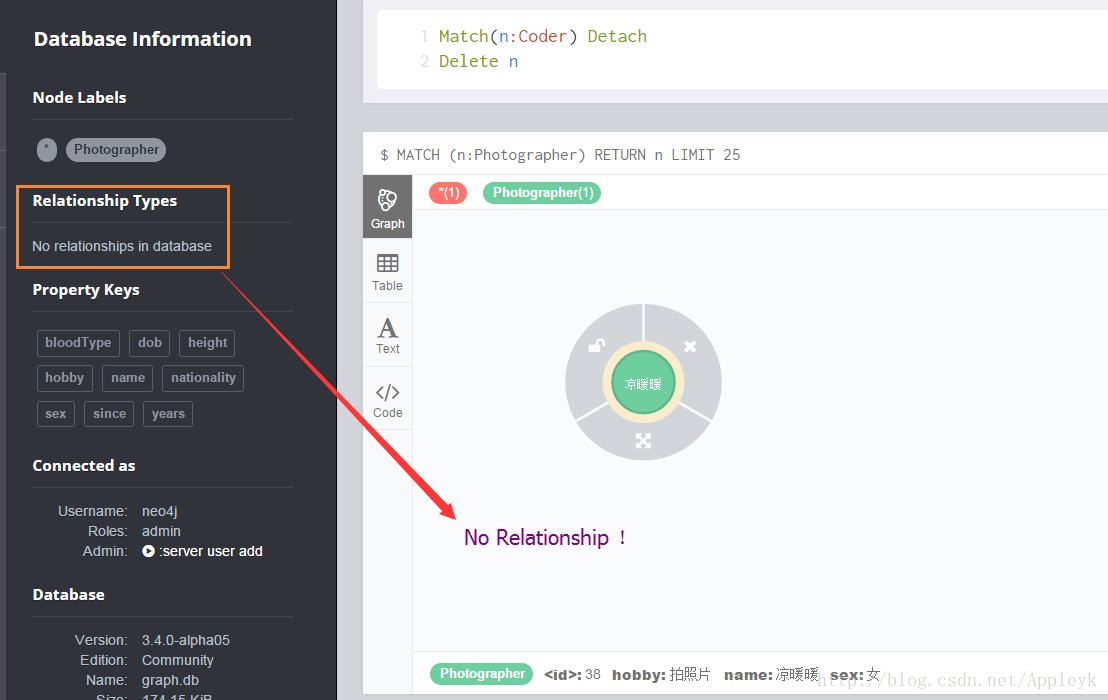
怎么办呢？ 匹配的时候加上一个Detach（分离）

Match(n:Coder) Detach  
Delete n



**意思就是，如果这个要删除的节点和其他节点存在关系的话，那么加上Detach就意味着接下来的删除将会是毁灭性的，也就是，不仅删除节点，还要删除关系**

最后看到的效果就是

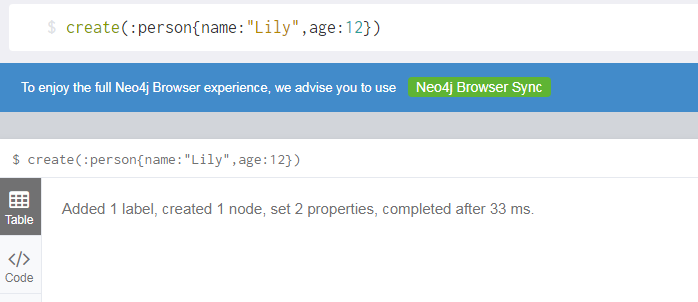


### 示例二:

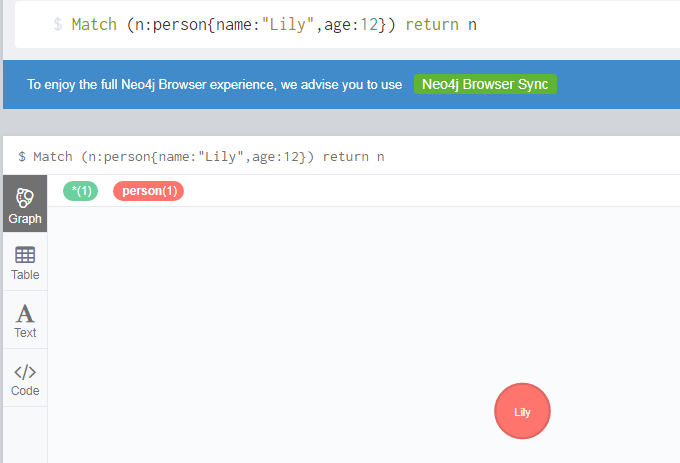
#### 单节点操作

##### 创建一个****节点****

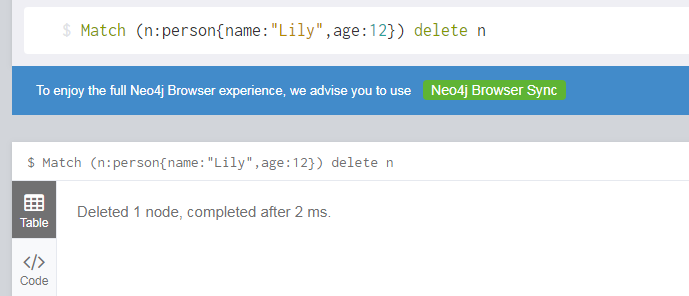
标签为Person的节点，其有属性 name 和age



##### 查询一个节点



##### 删除一个节点



#### Cypher官方文档

<https://neo4j.com/docs/developer-manual/3.4/cypher/>

#### 多节点操作

##### 创建三个节点

create(:Person{name:"a"}),(:Person{name:"b"}),(:Person{name:"c"})

##### 为其中的两个顶点添加边

            match (n:Person{name:"a"}),(m:Person{name:"b"})

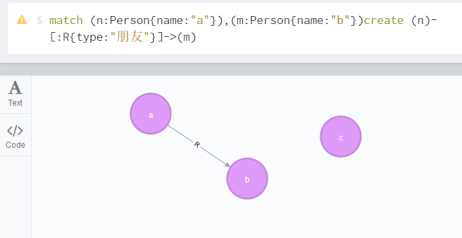
create (n)-[:R{type:"朋友"}]->(m)

           或：match(n),(m) where n.name=”a” and m.name=”b”

create (n)-[:R{type:"朋友"}]->(m)

           其基本思想是先找到要添加边的两个顶点，再为其创建边

上述两种方式是有缺陷的，因为有可能查出来不止一个节点，那么就会为许多匹配的节点都添加边



 上图出现了警告, 因此一般我们再创建新节点的时候就要创建边

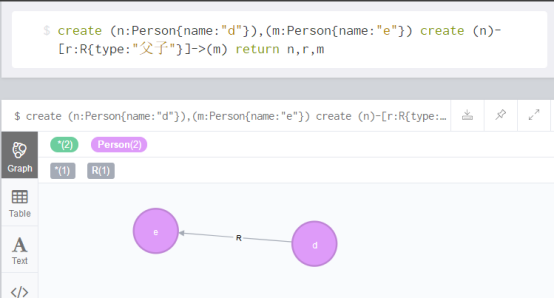
例如：

create (n:Person{name:"d"}),(m:Person{name:"e"})

create (n)-[:R{type:"父子"}]->(m)

             或：create (n:Person{name:"f"})-[r:R{type:"朋友"}]->(m:Person{name:"g"})

                使用 return 语句，返回创建结果视图



##### 删除关系（包含关系的查询）

match (n:Person{name:"f"})-[r:R{type:"朋友"}]->(m:Person{name:"g"}) delete r [该方式已过时不推荐使用]或

match p=(n:Person{name:"d"})-[:R\*1..2]->(m:Person{name:"e"})

delete p [推荐使用该语句]

说明：

第二条里面的\*1..2 查询的路径长度，最小为1，最大为2的关系为R的路径

在删除关系之前要先查询到该关系；

## ****基本的思想****

 Cypher 语句是Neo4j图形数据库的查询语言

   基本思想需要掌握的是：顶点的表示 （n）

                         关系（边）的表示 （n）-[r]->(m)

      这里的n r m可以看作是变量

   详细的语法可参考

<https://neo4j.com/docs/developer-manual/current/cypher/syntax/patterns/#cypher-pattern-varlength>

Cypher 内容丰富，与SQL 类似，同样包含诸多函数等；基本上能满足各种需求。因时间关系，未能一一学习了解

## ****Neo4j的使用方式****

1. 基于浏览器的可视化，使用Cypher语句进行相关操作
2. 基于Java 等编程语言的API 操作

## ****基于Neo4j Native Java API 的应用介绍****

### Git代码

https://github.com/EthonCao/neo4j.git

### Maven依赖

   若无可视化需求，可以在不安装neo4j server 的情况下使用Neo4j,只需要导入Neo4j的jar包即可

   <dependency>  
    <groupId>org.neo4j</groupId>  
    <artifactId>neo4j</artifactId>  
    <version>3.4.9</version>  
</dependency>