KADE知识图谱运维系统安装说明

目录

[一、数据库配置说明 2](#_Toc529328086)

[1）安装mongodb数据库 2](#_Toc529328087)

[2）启动mongodb服务 2](#_Toc529328088)

[3）安装mongodb GUI：Studio-3T 3](#_Toc529328089)

[4）通过studio-3T链接mongodb数据库 3](#_Toc529328090)

[5）导入数据库 3](#_Toc529328091)

[二、KADE知识图谱运维系统配置说明 6](#_Toc529328092)

[1）配置环境 6](#_Toc529328093)

[2）启动系统 7](#_Toc529328094)

[3）应用示例 9](#_Toc529328095)

[3.1.单表查询 9](#_Toc529328096)

[3.2关联查询 11](#_Toc529328097)

[3.3 批量导入 14](#_Toc529328098)

[3.4 配置数据库 17](#_Toc529328099)

# 一、数据库配置说明

此目录下需要的所有数据，均可在“graphfiles\数据库相关”下找到

## 1）安装mongodb数据库

运行 mongodb-win32-x86\_64-2008plus-ssl-3.6.0-signed.msi （适用于windows 7/10/server)

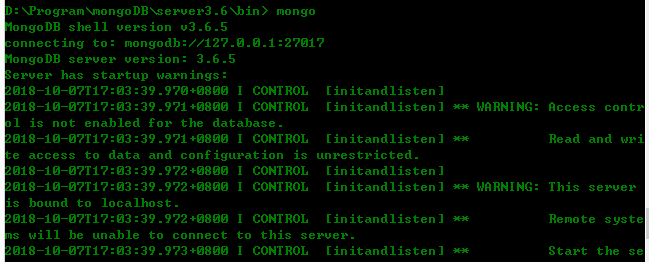
注意！！！安装过程中如果问你是否需要安装“Compass”软件时，不要安装！！！

判断mongoDB是否安装成功：

首先开启mongoDB服务（见（2））

新打开一个cmd，进入mongoDB安装目录，输入mongo,如果出现下面提示，则安装成功。

配置环境变量：path变量添加：D:\Program\mongoDB\server3.6\bin;



## 2）启动mongodb服务

2.1 在C盘创建data文件夹

2.2 在C:\data文件夹内创建db文件夹

2.3找到mongod.exe所在文件夹，在cmd命令行下运行下列语句

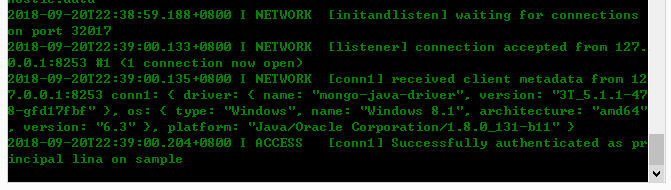
C:\Program Files\MongoDB\Server\3.6\bin> .\mongod.exe --dbpath=c:\data\db

2.4指定端口启动

Cmd进入mongoDB\server3.6\bin目录， 输入以下命令：

mongod --dbpath D:/Program/mongoDB/server3.6/data/db --port 32017（路径修改为mongoDB安装位置，端口可更改）





## 3）安装mongodb GUI：Studio-3T

运行 studio-3t-x64.msi

## 4）通过studio-3T链接mongodb数据库

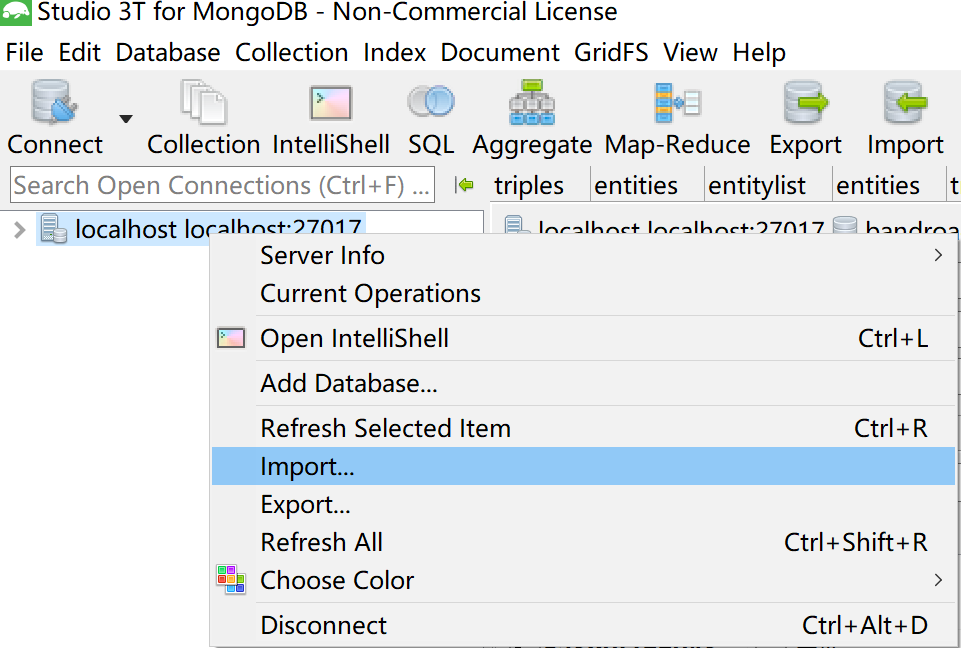
4.1 点击“Connect"按钮，新建一个链接”New Connection"，将name设置为localhost，其他不动，点击save保存。

4.2 连接localhost即可连接本地mongodb数据库

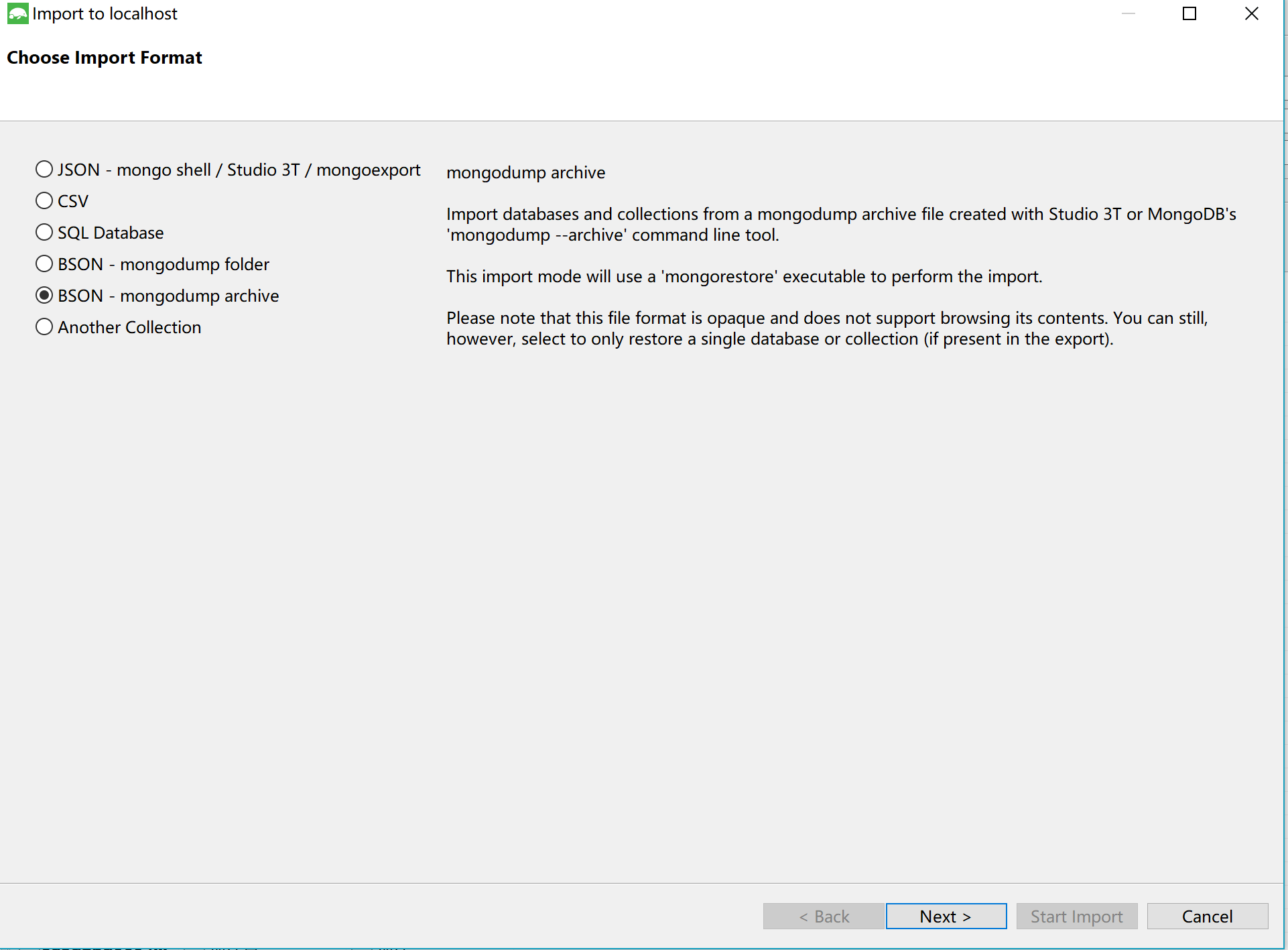
## 5）导入数据库

将graph\_kg.agz通过studio 3T软件导入到mongodb数据库中，具体如下：

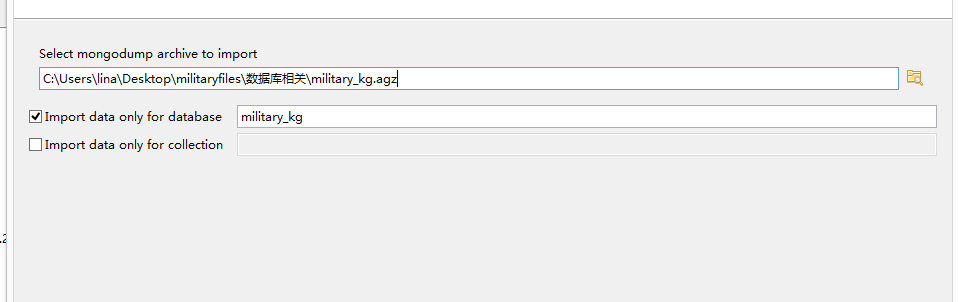
第一步，将鼠标移动到localhost数据库中，点击鼠标右键，点击“Import”



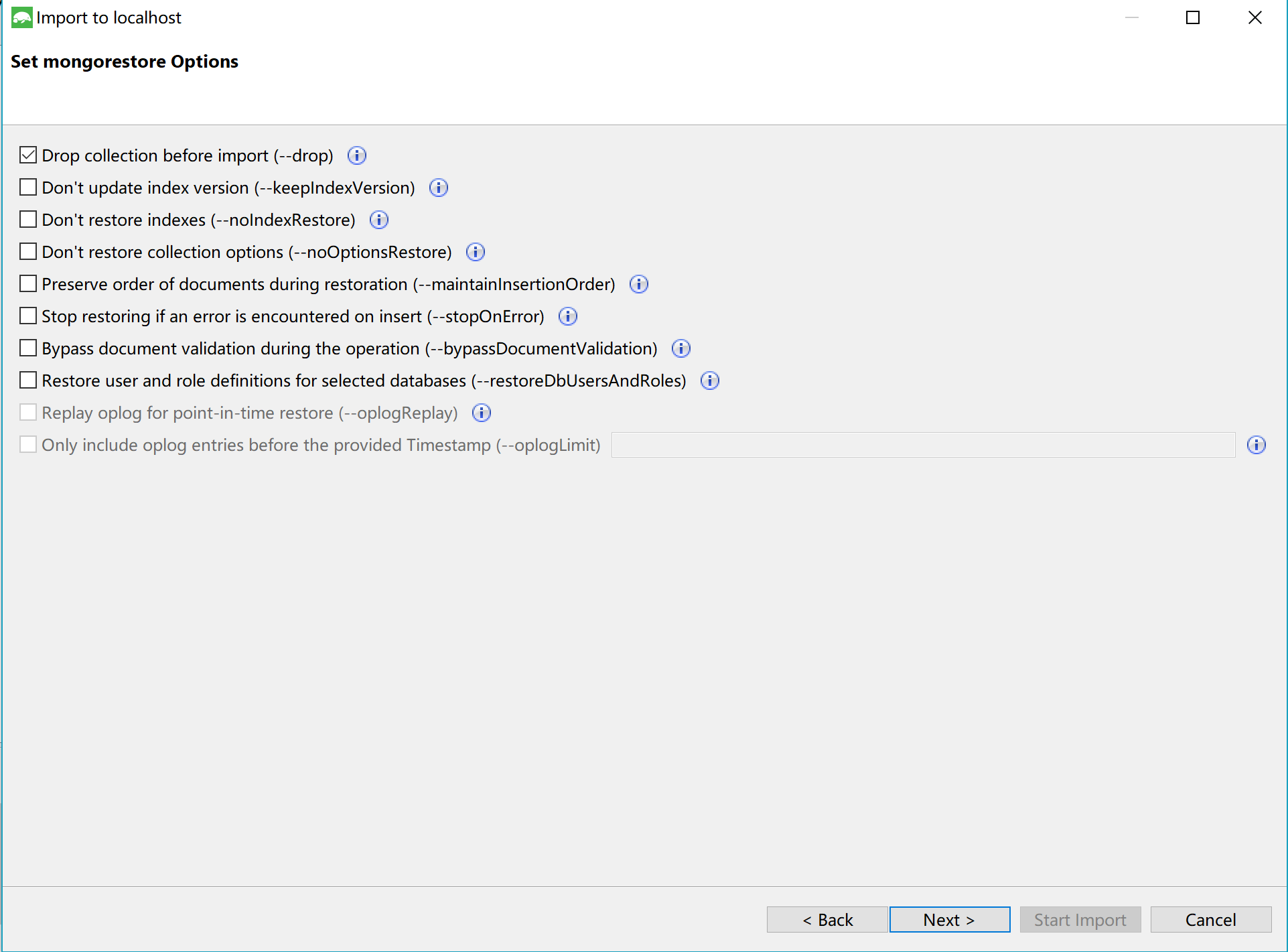
第二步：选择”BSON- mongodump archive”，点击“Next”



第三步，选择graph\_kg.agz所在地址，在Import data only for database上打钩，并输入新的数据库名。这里设置数据库名为“graph\_kg”,点击“Next”

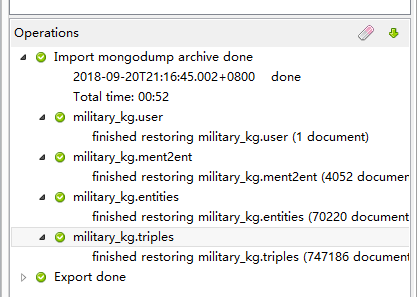


第四步，只在第一个选项上打钩，其他地方不用打钩。点击“Next”



第五步，点击“Start Import”

第六步，导入完成，屏幕左下角会告知导入完成



# 二、KADE知识图谱运维系统配置说明

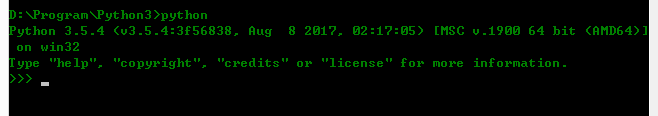
## 1）配置环境

第一步，安装Python

运行 python-3.6.4-amd64.exe

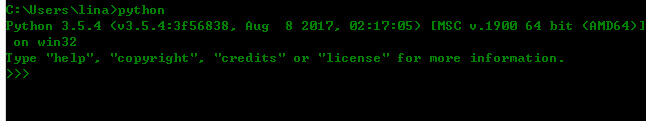
判断python3是否安装成功：

1.进入python安装目录，输入python，若出现以下内容，则安装成功。



2.配置完环境变量后，在任意目录下，输入python，若出现以下内容，则安装成功。

在path 中添加：D:\Program\Python3;D:\Program\Python3\Scripts;



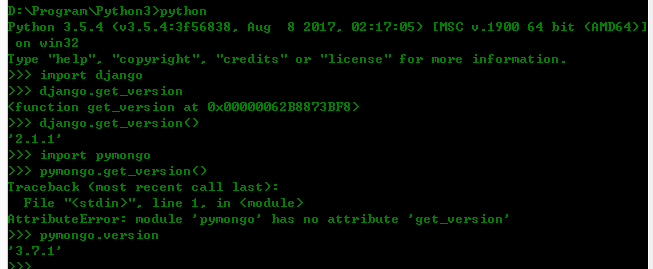
第二步，安装Python第三方库

|  |  |  |
| --- | --- | --- |
| 库名 | 安装方式 | 说明 |
| Django | 在线安装  pip install django | 微型Python Web 框架 |
| 离线安装  pip install Django-2.1.1.tar.gz | |
| pymongo | 在线安装  pip install pymongo | 驱动程序，使python程序能够使用Mongodb数据库 |
| 离线安装  pip install pymongo-3.7.1-cp36-cp36m-win\_amd64.whl | |

判断python第三方库是否安装成功：

进入python，输入import django , django.get\_version();

import pymongo , pymongo.version;若正常出现包的版本，则安装成功。

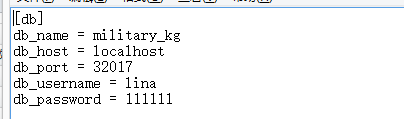


## 2）启动系统

进入“graph\kg”目录

第一步，修改数据库连接操作文件

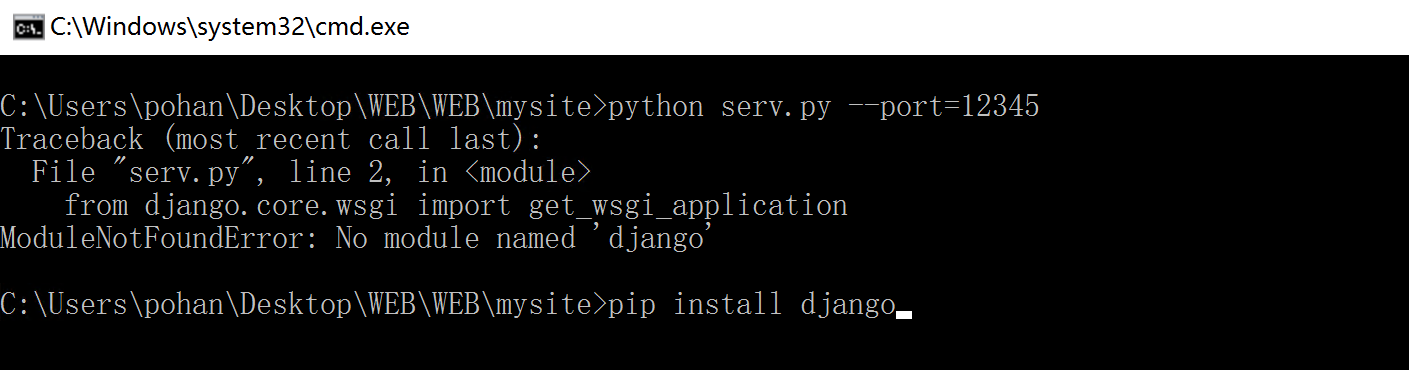
修改“graph\kg\db.ini”文件，修改文件中的以下参数（数据库认证的用户名密码为自己设置的）



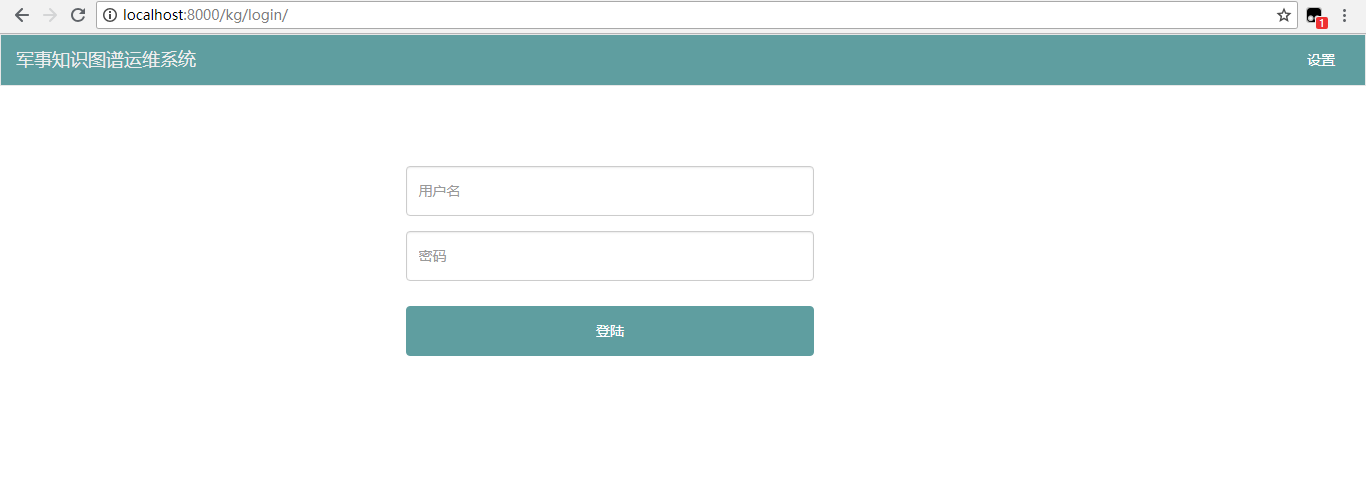
第二步，进入“graph\（与manage.py同级）”目录

运行cmd命令， python manage.py runserver 8000，注意端口可修改（不写端口为默认的8000）。

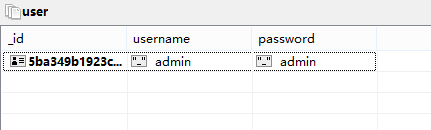
第三步，如果提示有些第三方库为安装，使用pip install \*\*进行安装



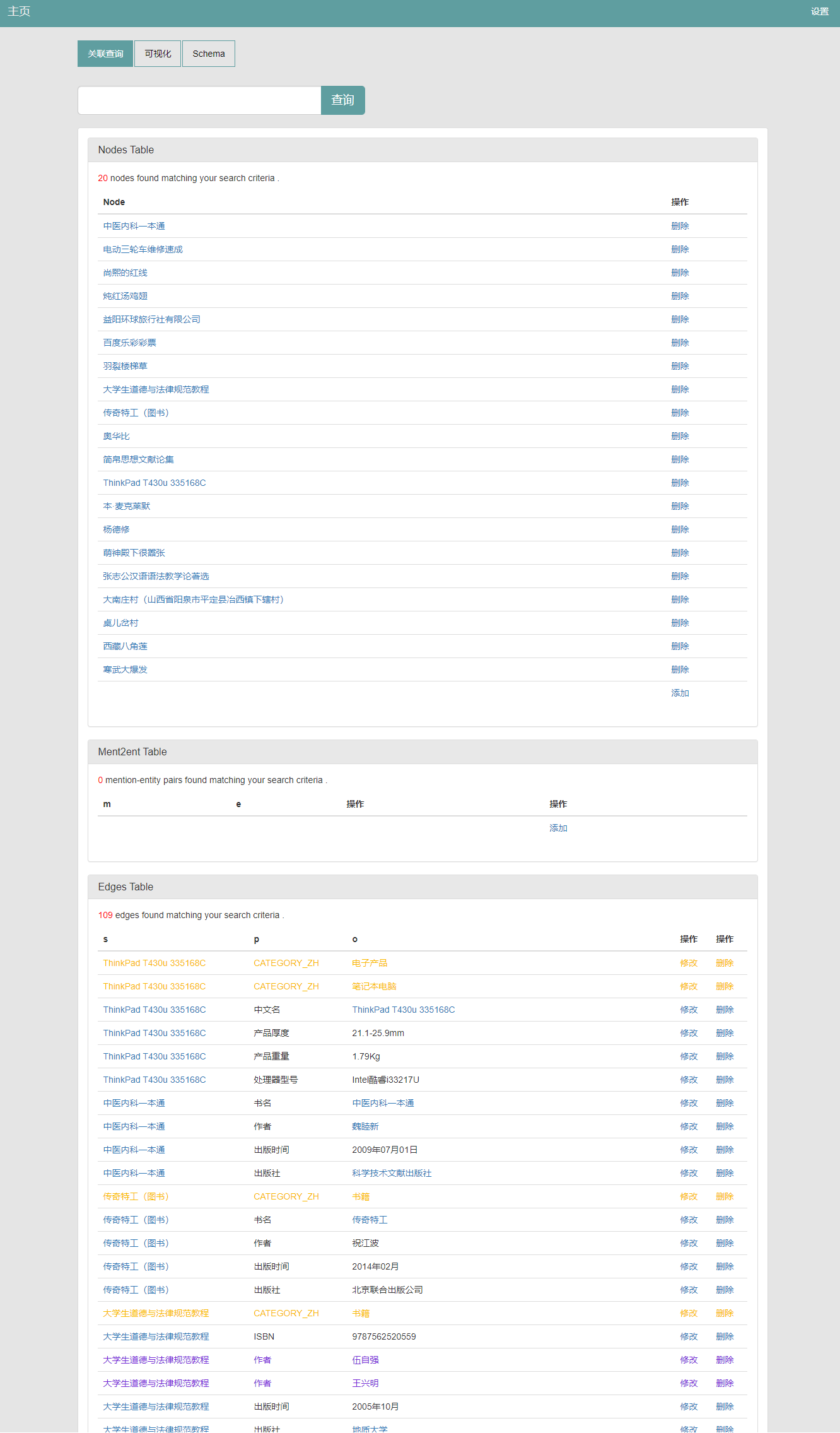
第四步，打开浏览器，输入http://localhost:8000/kg/login/，进入登录界面：



输入user表中的某个用户名、密码，即可登陆成功



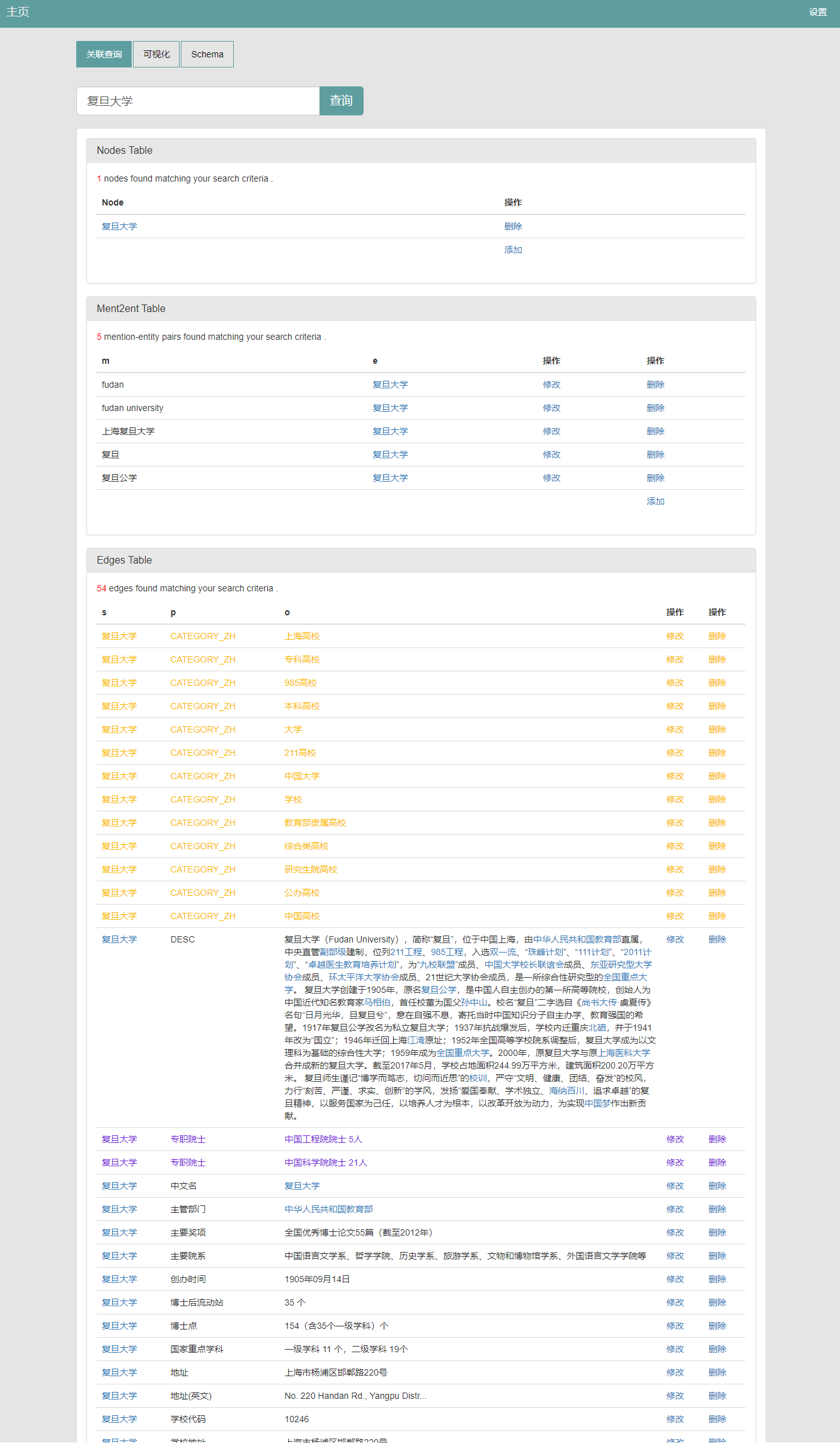
系统包括四个模块，关联查询、可视化编辑、schema编辑、数据库配置。



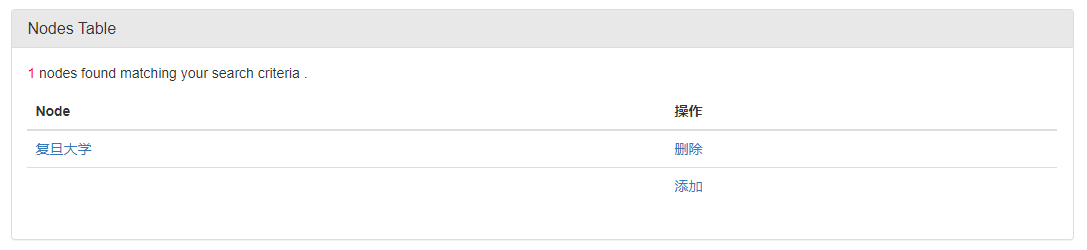
## 3）应用示例

### 3.1关联查询

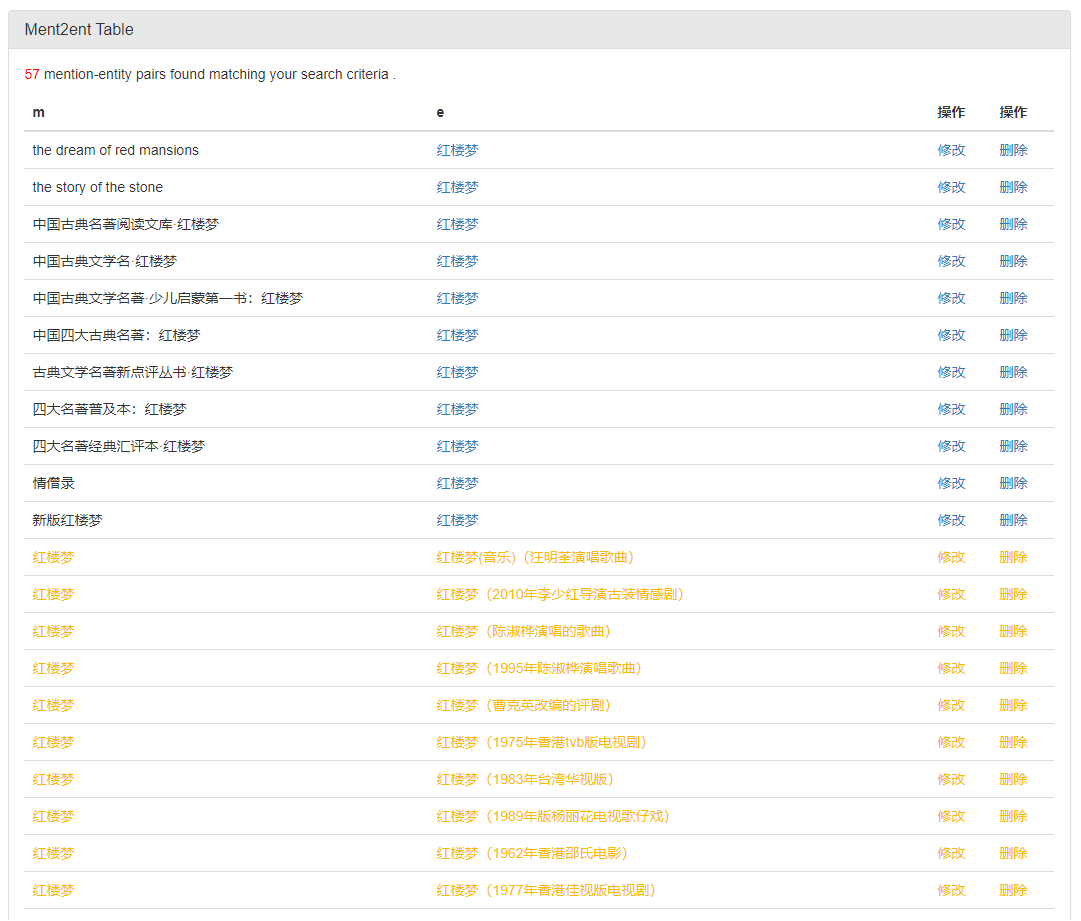
分别会显示当前所查询的实体、(mention,entity)pair、以及实体相关的三元组。



3.1.1 实体显示，支持增删实体。修改实体会导致会破坏其他数据的关联性，因此不提供此功能。



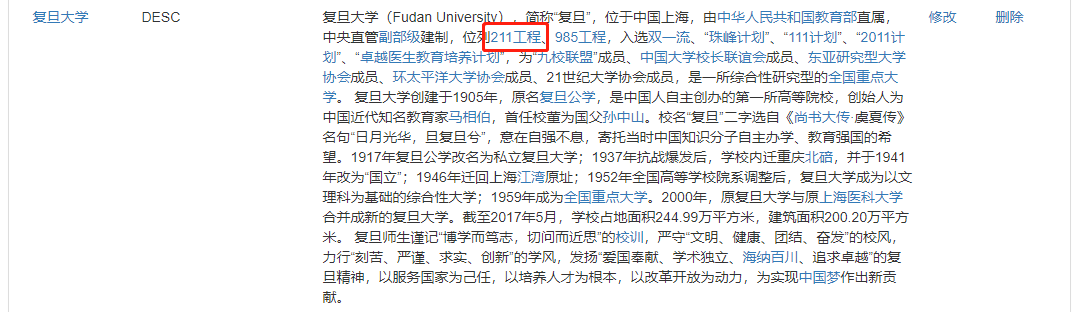
3.1.2 提及-实体对（mention,entity pair）显示。包括以搜索关键字为实体、以搜索关键字为提及（mention）的两部分内容。支持增删改查。



3.1.3 三元组显示。按属性分组显示，相同属性的三元组会分为一组，不同分组以不同的颜色区分显示。支持增删改查。



内容里包含超链接可支持跳转。

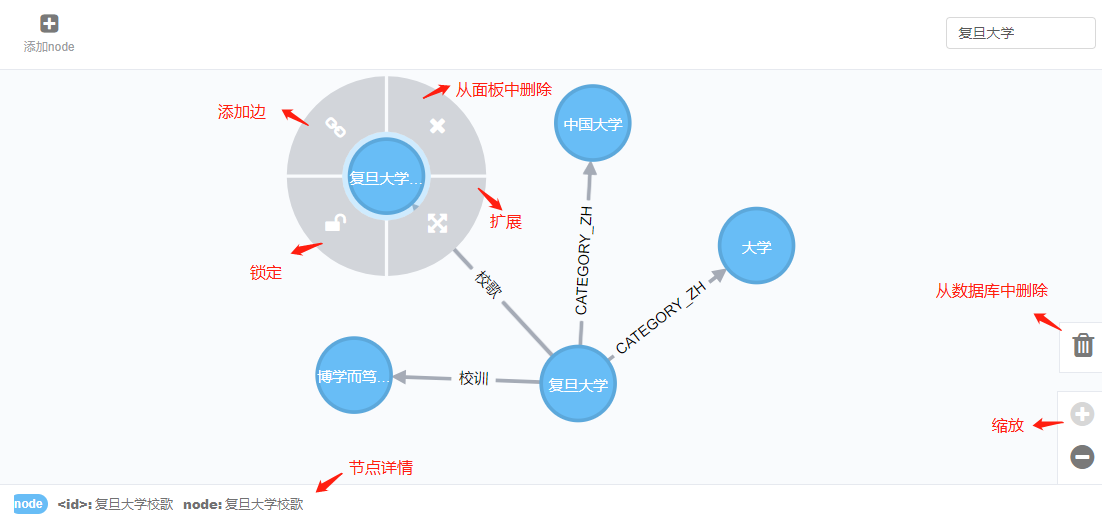


点击超链接跳转。

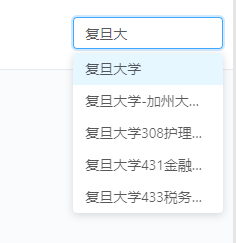


### 3.2 可视化编辑

#### 3.2.1 功能概览

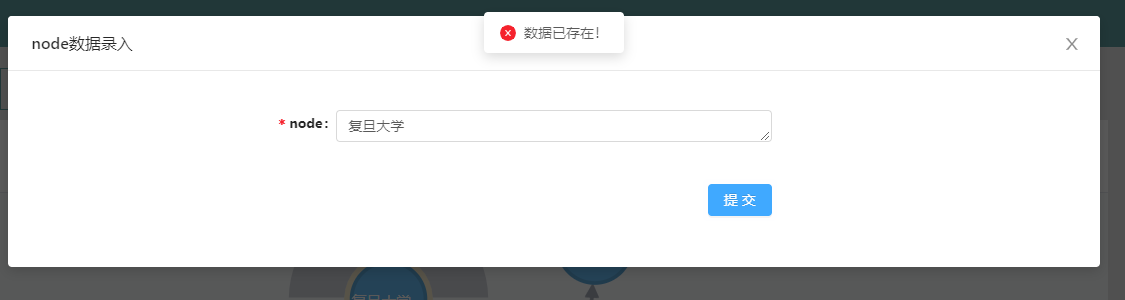


#### 3.2.2 查询建议



#### 3.2.3添加节点

数据不合法时会有错误提示。

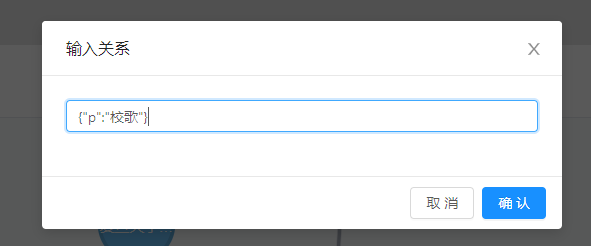


#### 3.2.4 添加边

选择目标节点后填写json格式的属性信息。

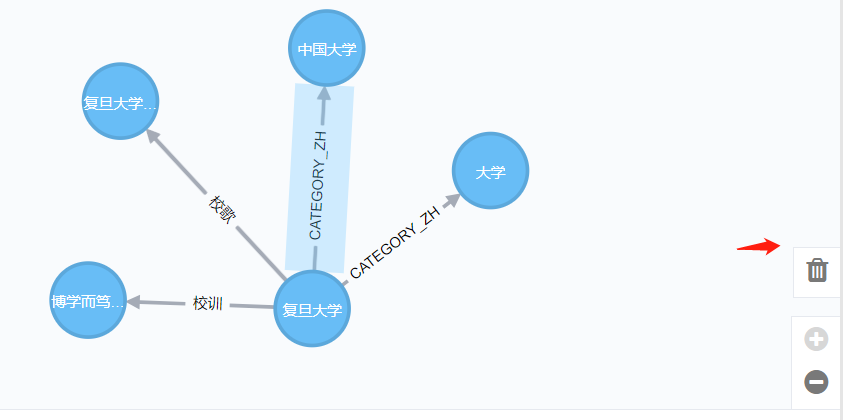
e.g 1 {“p”:”校歌”}

e.g 2 {“w”:0.98,”tp”:”coocc”}



#### 3.2.5 删除边

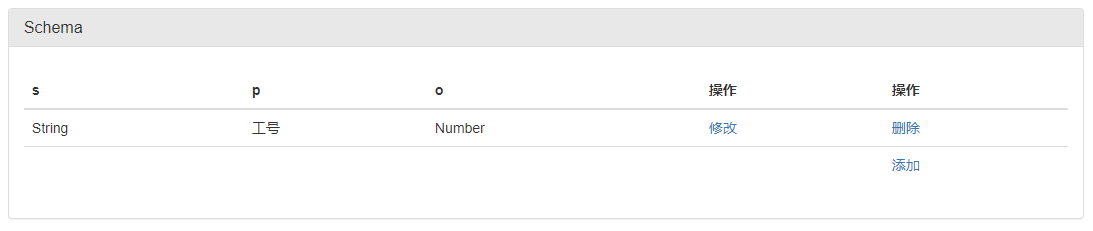
选中边，点击删除按钮即可。



### 3.3 配置Schema

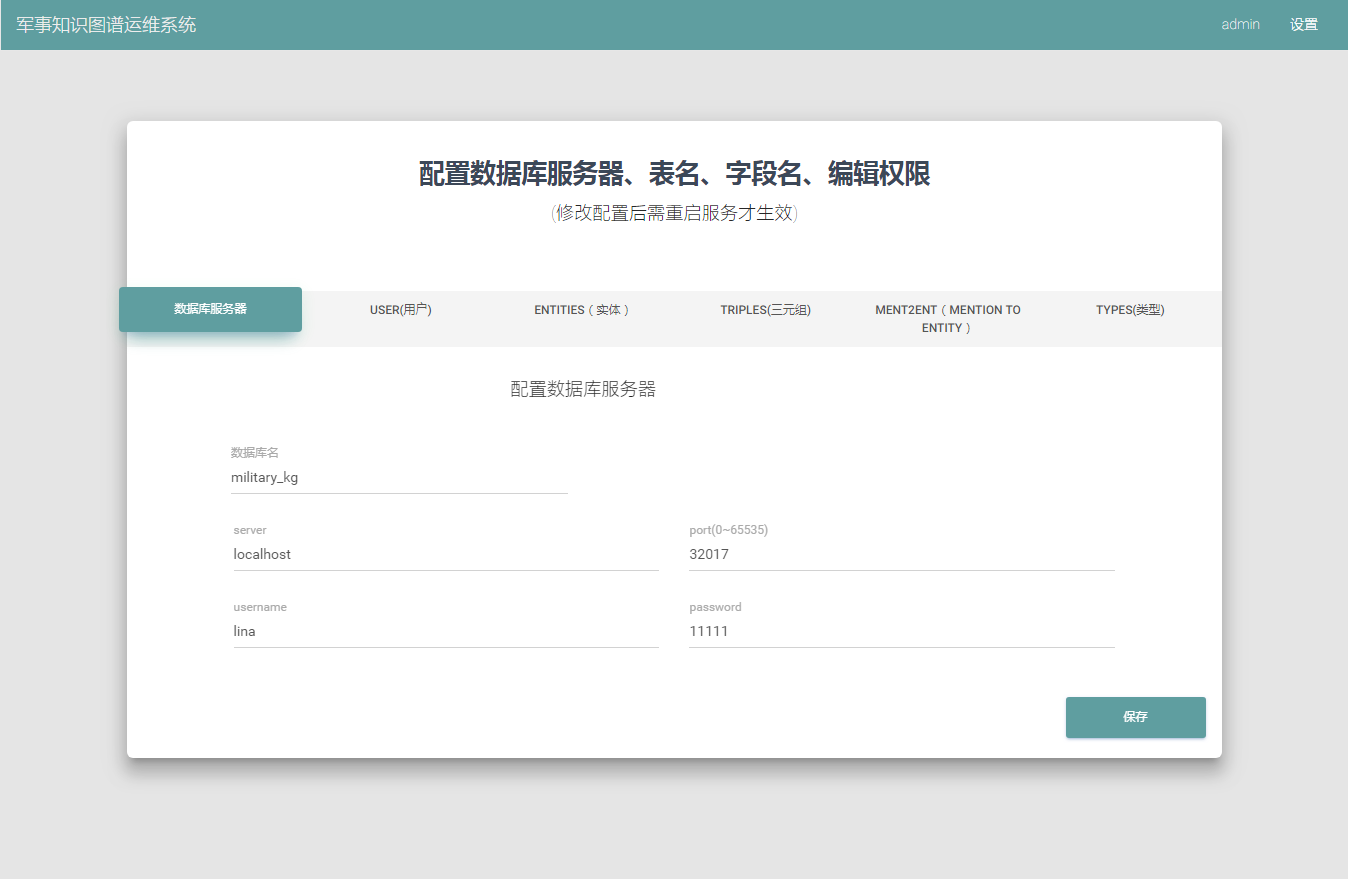
建立对于三元组数据的约束。

如下约束的含义为：若新增三元组的属性（p）为“工号”，则需要满足subject为字符串，object为数字，不满足schema约束的三元组无法插入。



### 3.4 配置数据库

配置数据库服务器。配置对应于graph\kg\db.ini配置文件中的内容。此模块做的修改需重启服务才生效。



## 4）图谱结构配置

本系统适用于**通用模式**，只需要修改配置文件便可做到**一套系统同时运维任意结构（不同表名、不同字段名、不同结构）的图谱**。

以下是修改配置的两种不同于以上示例结构的运维系统。

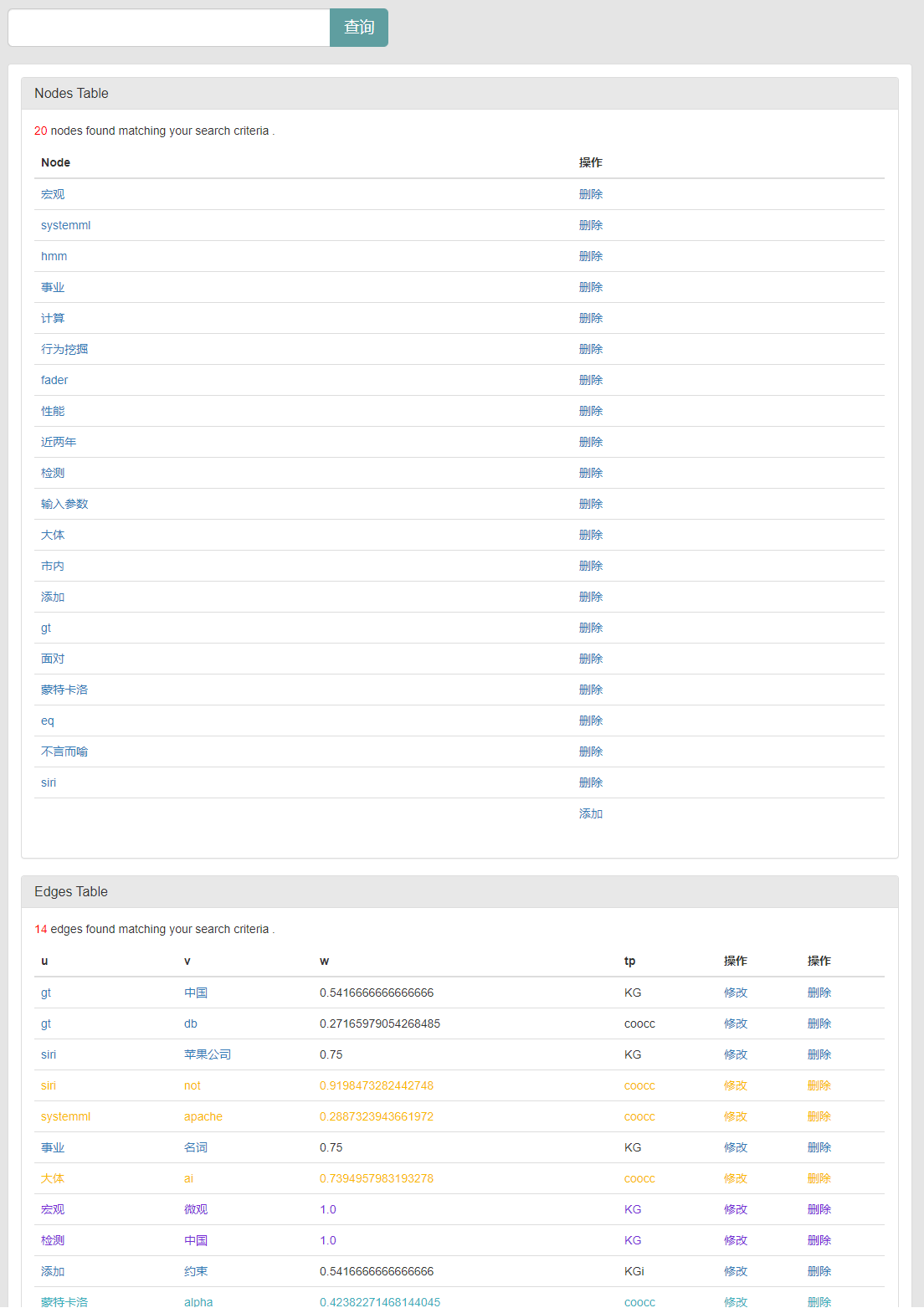
e.g 1

spo三元组中的object包含实体，将object拆分为 object mention、object entity分开显示。

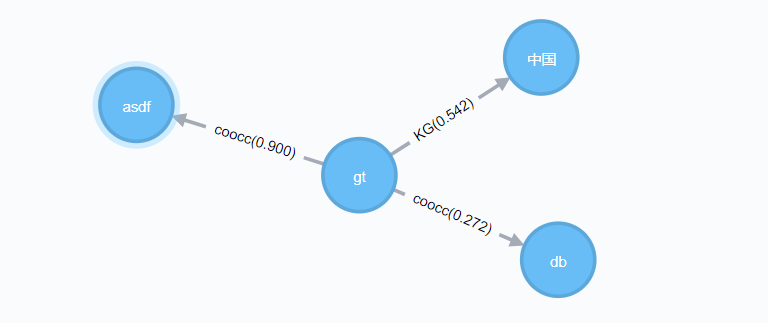


e.g 2

这类图谱包含两张表，三元组中的属性有两个内容——w(权重)、tp(类型)。







4.1 配置文件说明

参见args.py,args\_.py 修改好配置文件后，再修改api.py中调用配置文件的代码，换成相应的配置文件名（e.g args\_） from . import **args\_** as args

1. **TRIPLE\_ARG**

三元组配置，json格式，key为表的字段名，key对应的value也是一个json,包含两个属性——node和sort。

(1)node表示此属性是否是实体，用来控制点击此实体进行页面跳转。node的取值为{0,1,2}。0：非实体/包含实体 1：全为实体 2： <a href=”oe”>om</a>格式。

(2)sort表示是否为排序关键字，取值为{0,1},所有key中只能有一个sort的值为1。

**e.g**

TRIPLE\_ARG={ 's':{'node':1,'sort':0}, 'p':{'node':0,'sort':1}, 'o':{'node':0,'sort':0} }

1. **MENT2ENT\_ARG**

与TRIPLE\_ARG配置相同。若没有ment2ent表，则将MENT2ENT\_ARG设为None。 **e.g** MENT2ENT\_ARG={ 'm':{'node':0,'sort':1}, 'e':{'node':1,'sort':0} }

1. **XXX\_COL**

把表的字段名按顺序排成一个字符串，若有字段名不是单字符，则把字段名排成字符串数组。 **e.g 1** TRIPLE\_COL = 'spo' MENT2ENT\_COL = 'me'

**e.g 2**

TRIPLE\_COL = ['u', 'v', 'w', 'tp']

1. **TRIPLE\_S、TRIPLE\_O**

指定三元组中两个实体分别在TRIPLE\_COL中对应的索引。 TRIPLE\_S = 0 TRIPLE\_O = 2

1. **def PRINT\_PREDICATE(edge):**

定义打印三元组predicate（谓语）的函数，即可视化查询时两个实体之间边的显示格式。

**e.g 1**

def PRINT\_PREDICATE(edge): return edge['p']

**e.g 2**

def PRINT\_PREDICATE(edge): return '%s(%.3f)' % (edge['tp'], float(edge['w']))

1. **ENTITY\_ARG**

实体表配置，包含一个属性entity\_name，值为实体表中实体字段的属性。

**e.g**

ENTITY\_ARG={ 'entity\_name':'\_id' }

1. **ENTITIES\_USE\_OBJECTID**

表示实体表中的'\_id'字段是否为OBJECTID。若'\_id'字段为字符串则设为False,这种设计是为了方便建索引。 ENTITIES\_USE\_OBJECTID = False

1. **TABLENAME\_ARG**

表名配置。分别对应用户表、三元组表、实体表、ment2ent表、schema表。

**e.g**

TABLENAME\_ARG={ 'usertablename' : 'user', 'edgestablename' : 'triples', 'nodestablename' : 'entities', 'ment2enttablename':'ment2ent', 'schematablename':'schema1' }

1. **USER\_ARG**

用户表配置，包括用户名、密码。

**e.g**

USER\_ARG={ 'user\_username' : 'username', 'user\_password' : 'password' }

1. **SCHEMA\_COL**

把三元组表的字段名按顺序排成一个字符串，若有字段名不是单字符，则把字段名排成字符串数组。用来验证schema。 **e.g 1** SCHEMA\_COL='spo'

**e.g 2**

SCHEMA\_COL=['u', 'v', 'w', 'tp']