哈尔滨工业大学

**<<大数据分析>>**

**实验报告三**

**(2022年度春季学期)**

|  |  |
| --- | --- |
| **姓名：** | **李金宣** |
| **学号：** | **1181910201** |
| **学院：** | **大数据** |
| **教师：** | **张开旗 杨东华** |

实验三 图数据分析

## 实验目的

通过安装使用Neo4j以及利用Cypher对其进行基本

的增删查改；

加深对于图数据库和图模型的理解，学习使用图

数据库进行数据分析。

## 实验环境

操作系统：Windows

软件：Neo4j Desktop-1.3.4

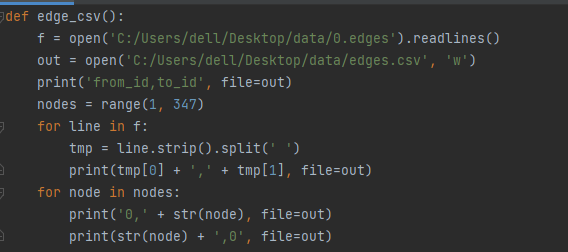
查询语言：Cypher

## 实验过程及结果

数据集的导入：

数据预处理：

0.edges处理为edges.csv，主要是标注from\_id和to\_id即边的顶点两个列



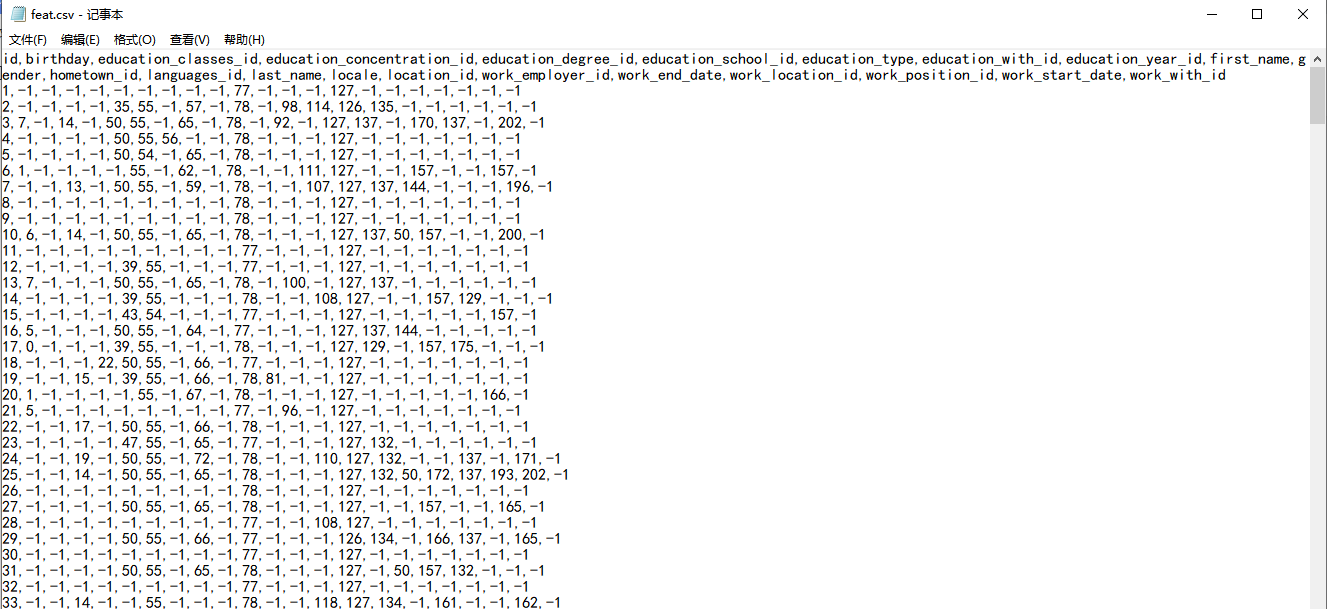
预处理效果：



0.featnames和0.feat处理为feat.csv，数据处理过程比较复杂，首先将0.featnames文件中每个特征的名称读入，存储到字典里，因为采取了one-hot的形式，所以每个特征名在字典中都对应若干长度的01队列，然后读取0.feat里的01特征序列，然后按顺序对应到每个特征，通过one-hot编码找对每个特征名对应的特征值，最后将特征名作为第一行的列名，其中第一列为，id写入feat.csv，将特征值写如每行顶点对应的列，若特征值不存在设为-1。



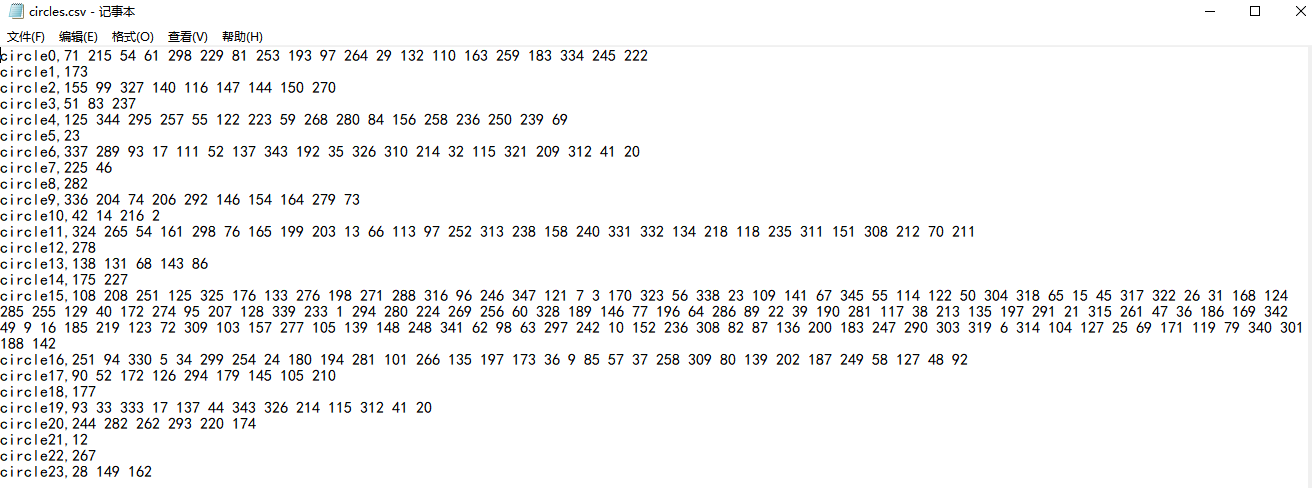
预处理效果：



0.circle到circle.csv，简单处理下格式即可



预处理效果：

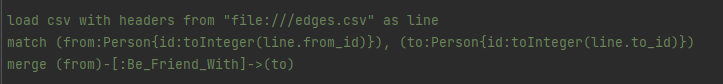


数据导入：

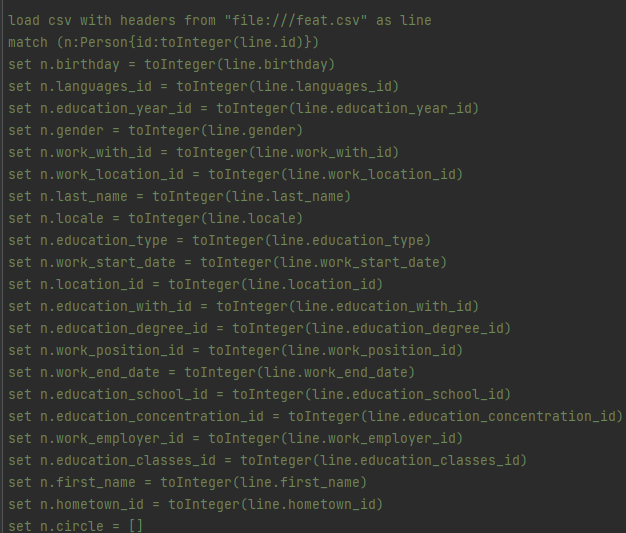
每个nodeID导入为一个节点，labelname为Person



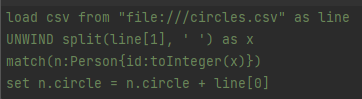
edges文件中描述的边（包括中心节点指向于每个节点的边）应当导入为关系，关系的label-name 为Be\_Friend\_With



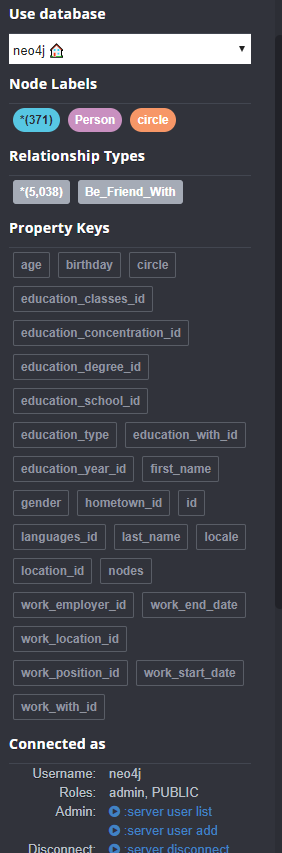
每个nodeId的feature应当被导入为属性



将circle导入，形式为对每个Person构造一个Circle列表，将其所属的circle加入其中



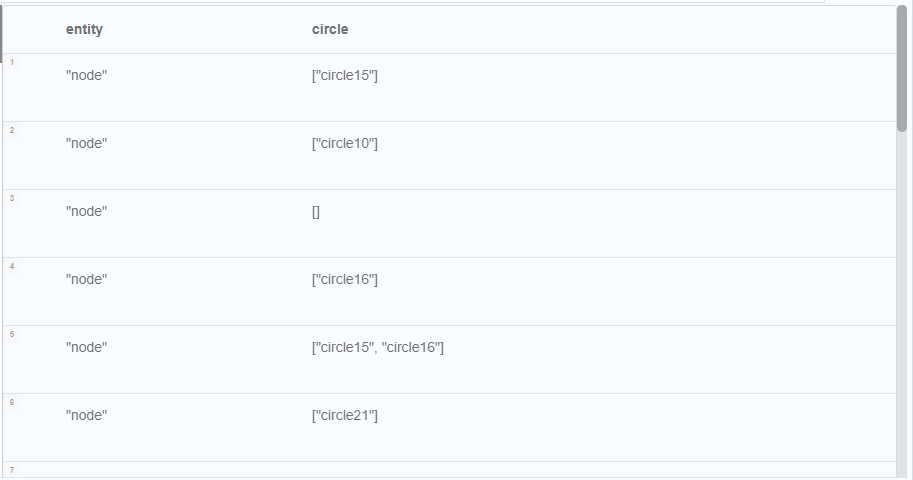
导入结果一览：



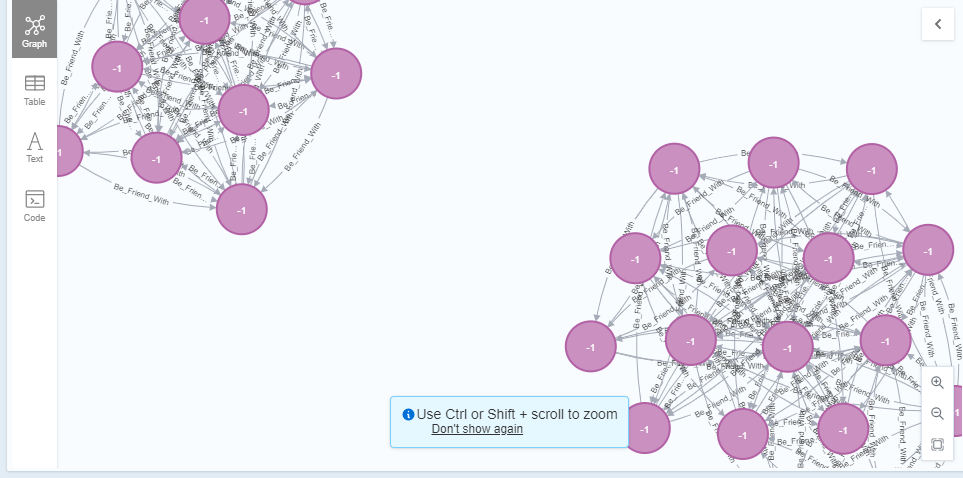
每个节点的ID



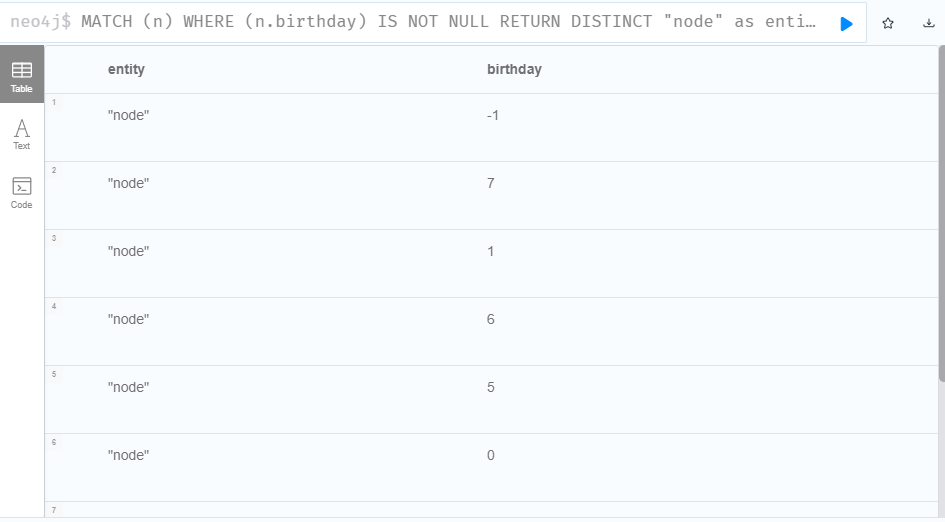
每个节点的circle



节点间朋友关系：

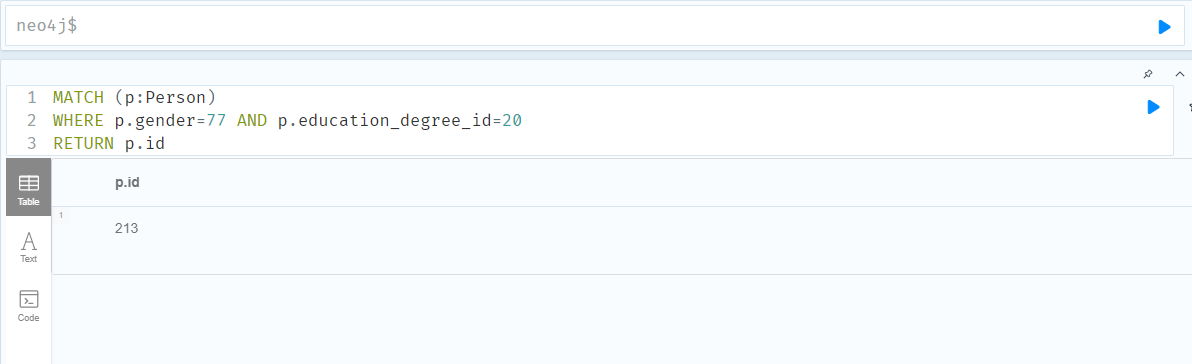


一些特征，如birthday

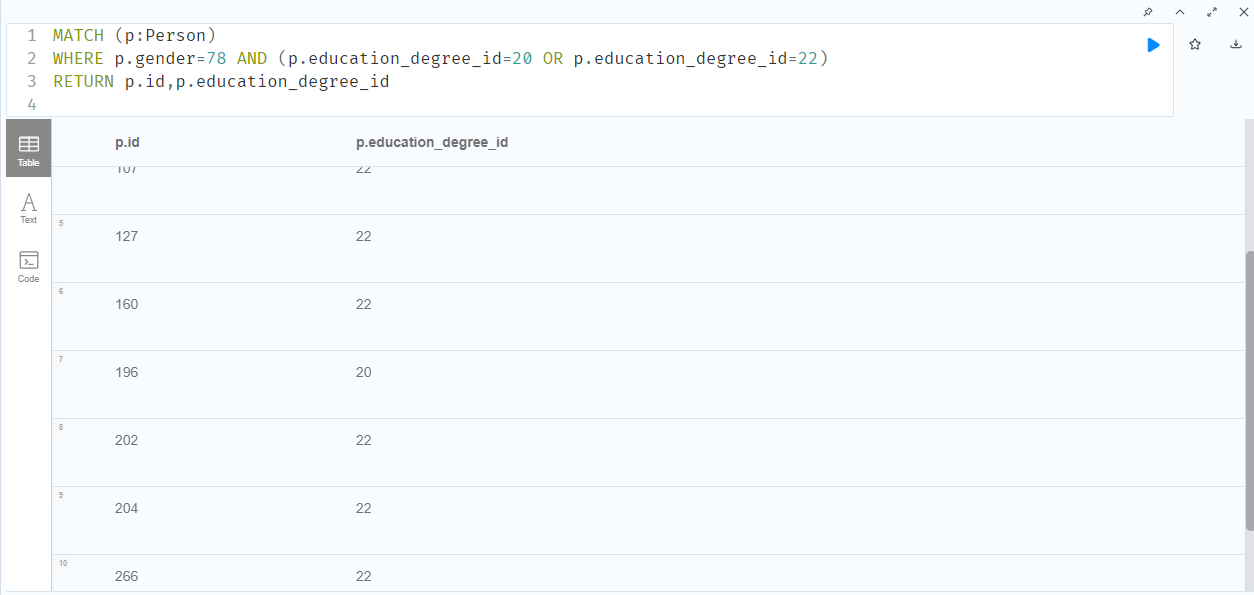


数据的查询：

1. 检索所有gender属性为77且education;degree;id 为20的Person；



1. 检索所有gender属性为78且education;degree;id为20或22的Person；



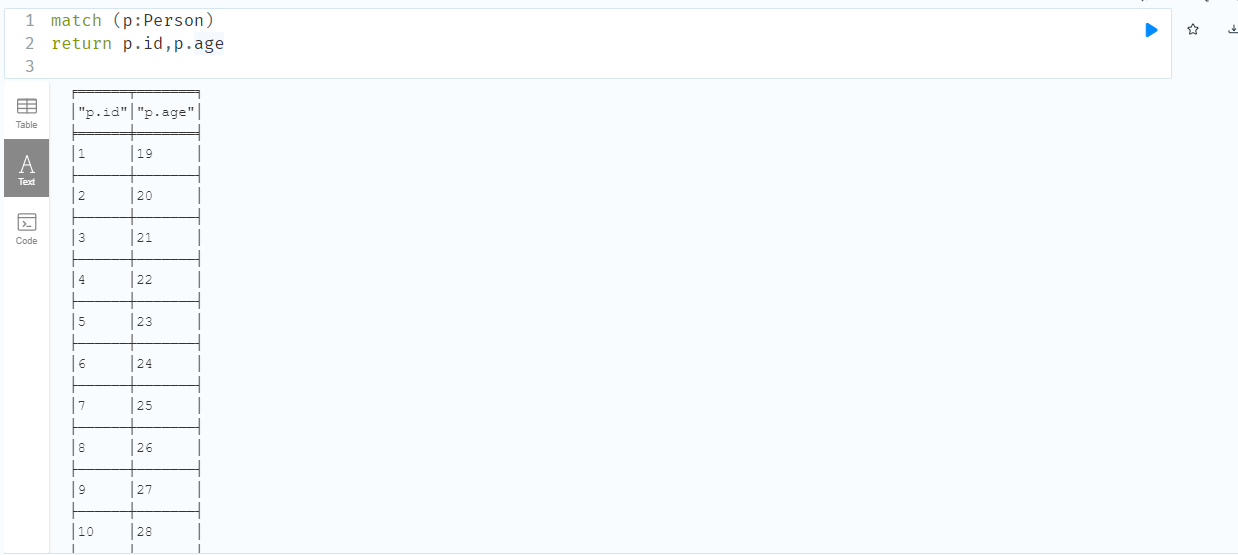
1. 为Person增设年龄age属性，数值自行设定，可以随机化，要求年龄介于18岁-30岁之间，尽量分布均匀

增设语句：id对13取余+18，age在18-30均匀分布

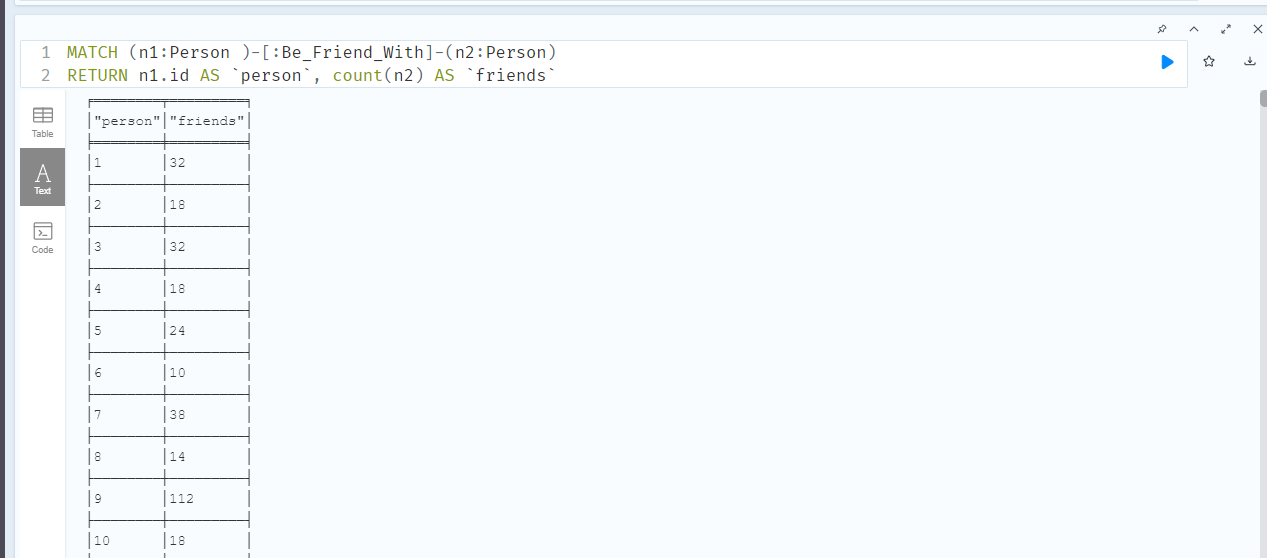
match (n:Person)

set n.age = n.id%13+18

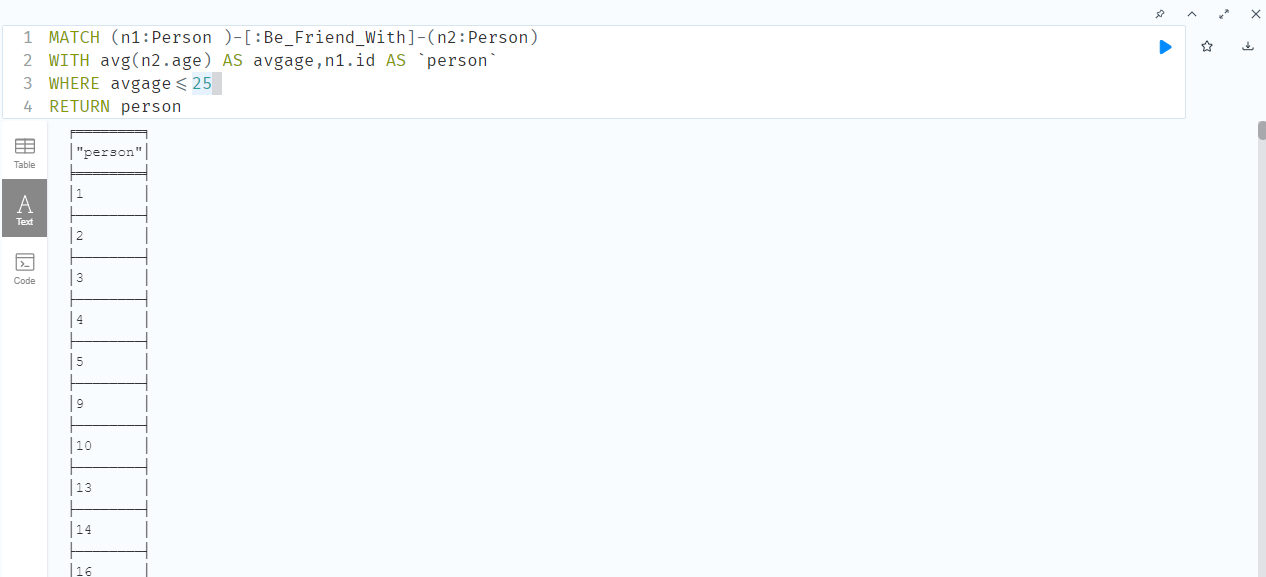
查询结果：



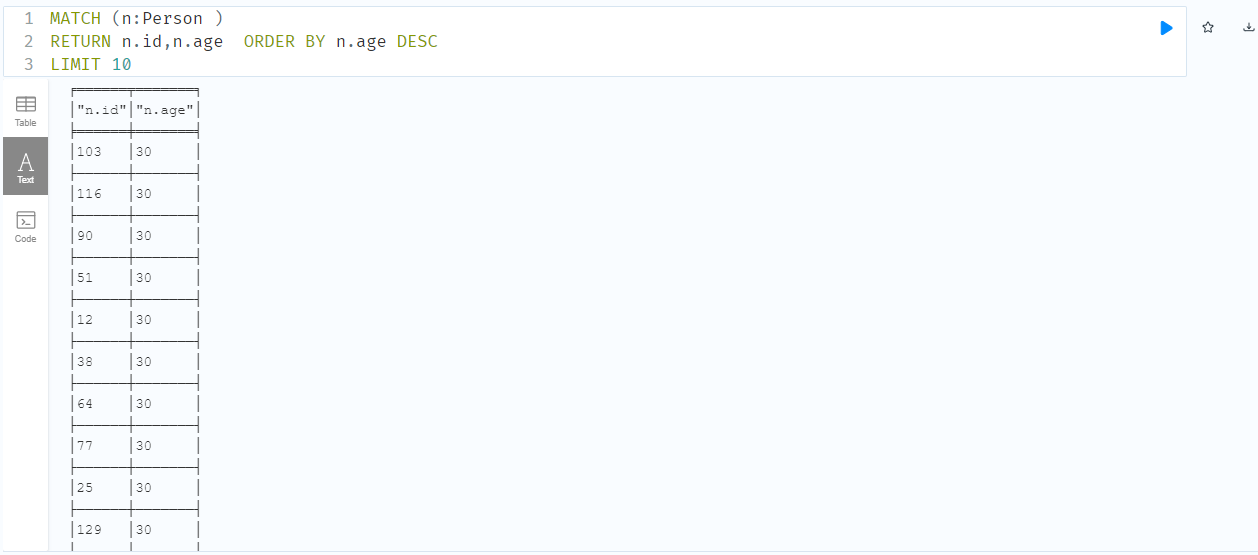
1. 检索每个Person的朋友的数量



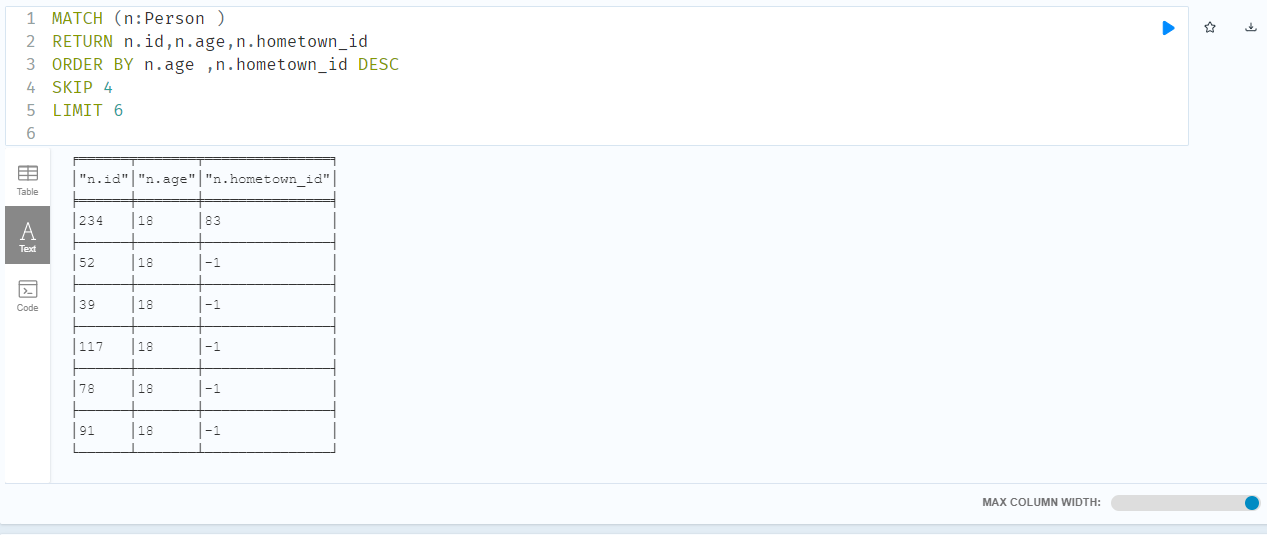
1. 检索朋友平均年龄值在25岁以下的Person集合



1. 检索年龄最大的前10个Person

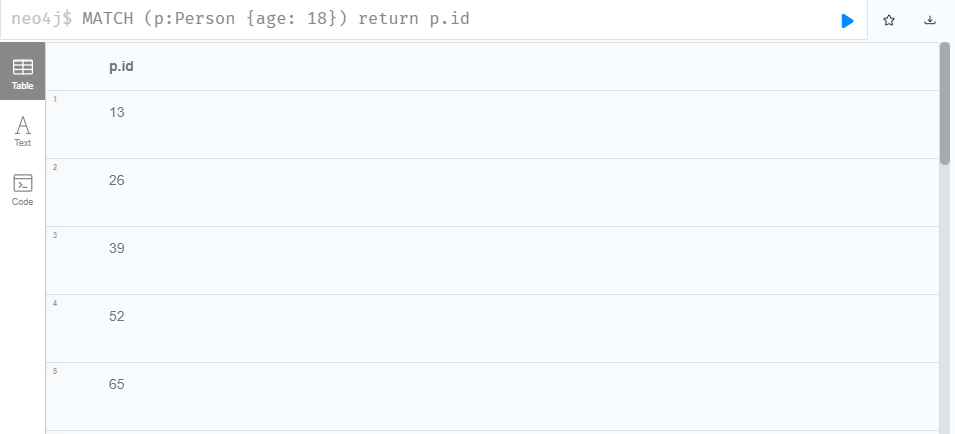


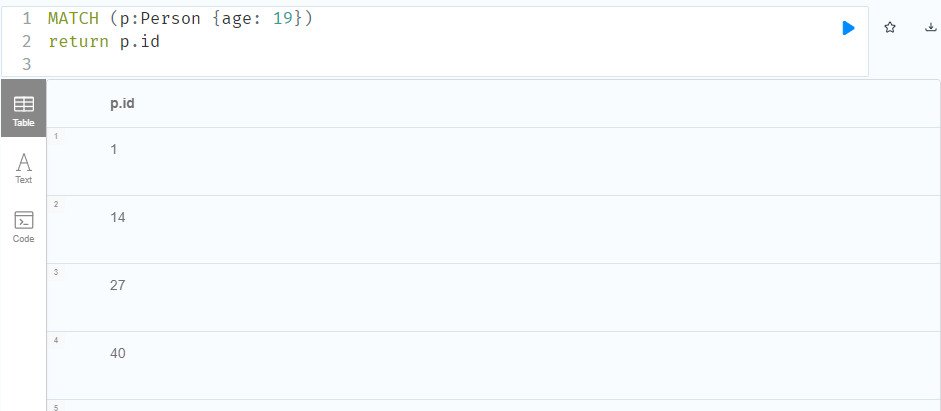
1. 按年龄排序(升序)所有Person后，再按hometown; id属性的字符串值(降序)排序，然后返回第5、6、7、8、9、10名Person



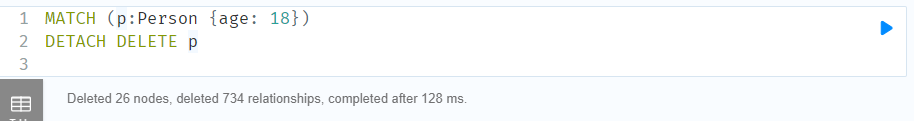
1. 删除所有年龄为18和19的Person

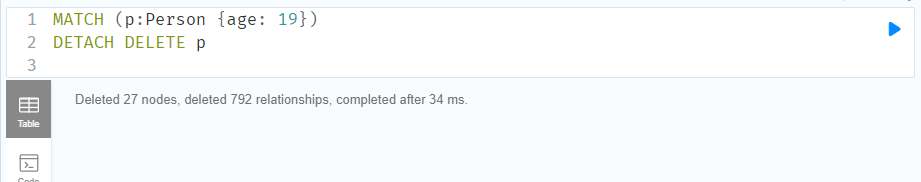
删除前：

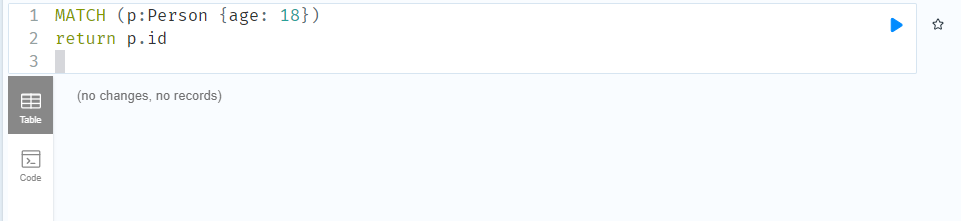


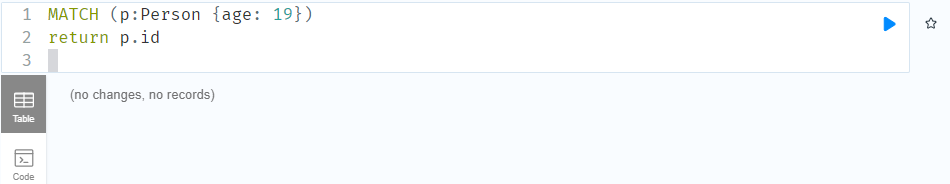


删除后，无查询结果









1. 对于人数少于两个的circle，删除掉这些circle里的Person的表示circle信息的属性

将circles.csv导入，已circle为节点，circle的nodes队列加入顶点点id

load csv from "file:///circles.csv" as line

create (c:circle{id:line[0]})

load csv from "file:///circles.csv" as line

UNWIND split(line[1], ' ') as x

match(c:circle{id:line[0]})

set c.nodes =[]

load csv from "file:///circles.csv" as line

match(c:circle{id:line[0]})

set c.nodes = c.nodes + line[1]

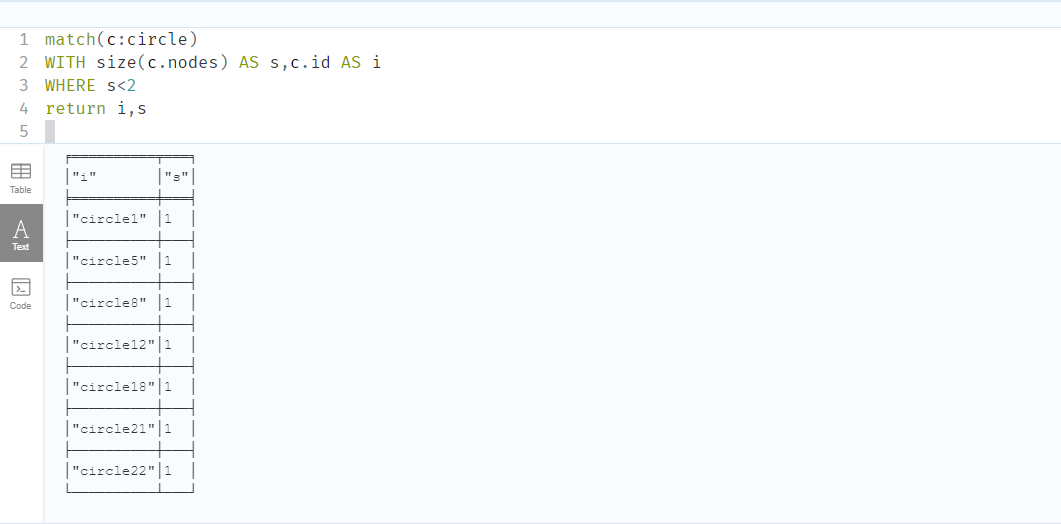
load csv from "file:///circles.csv" as line

UNWIND split(line[1], ' ') as x

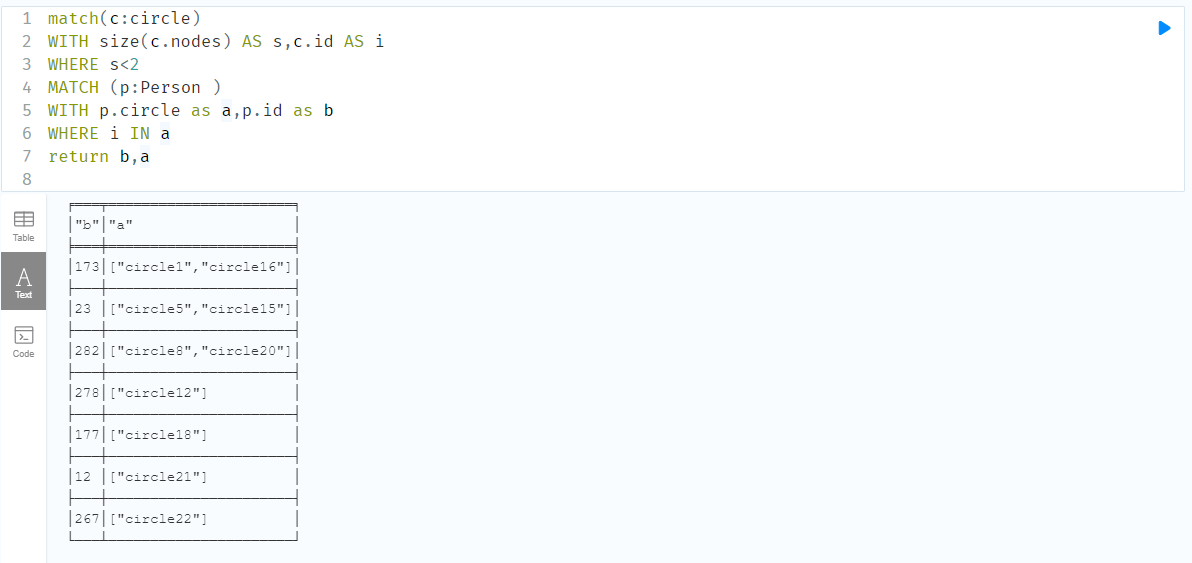
match(c:circle{id:line[0]})

set c.nodes =c.nodes+ toString(x)

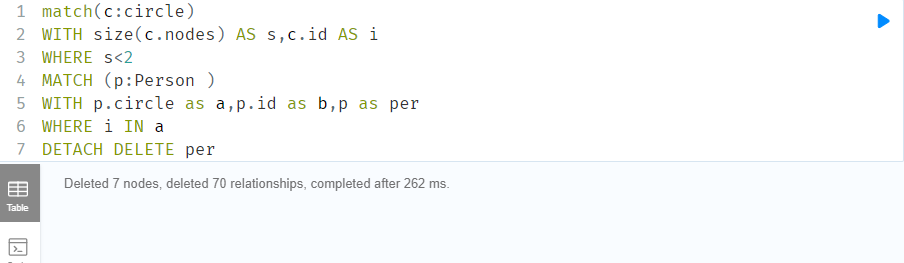
人数少于2个的circle



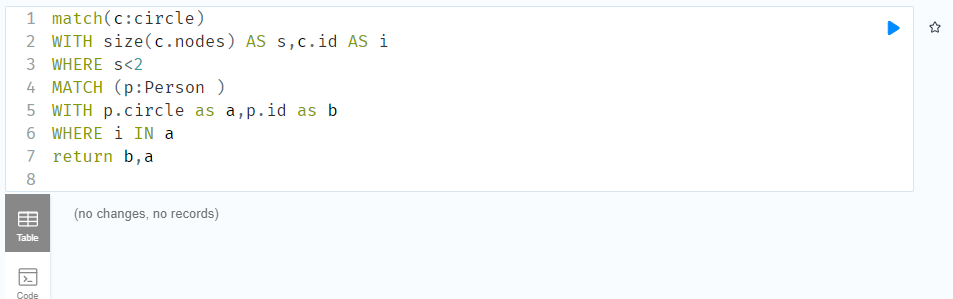
人数少于2个的circle里的person及表示circle信息的属性



删除操作



删除后：

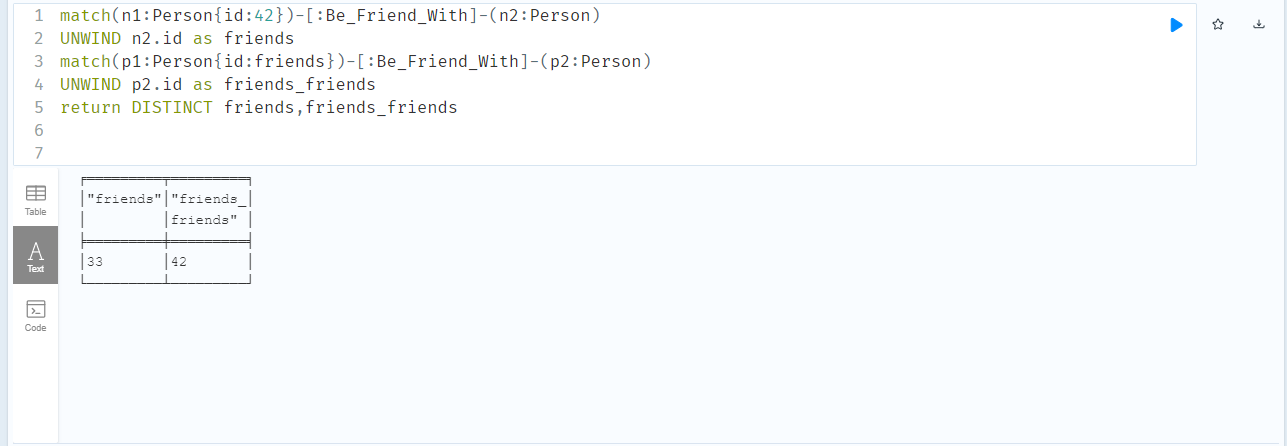


1. 检索某个Person的所有朋友和这些朋友的所有朋友

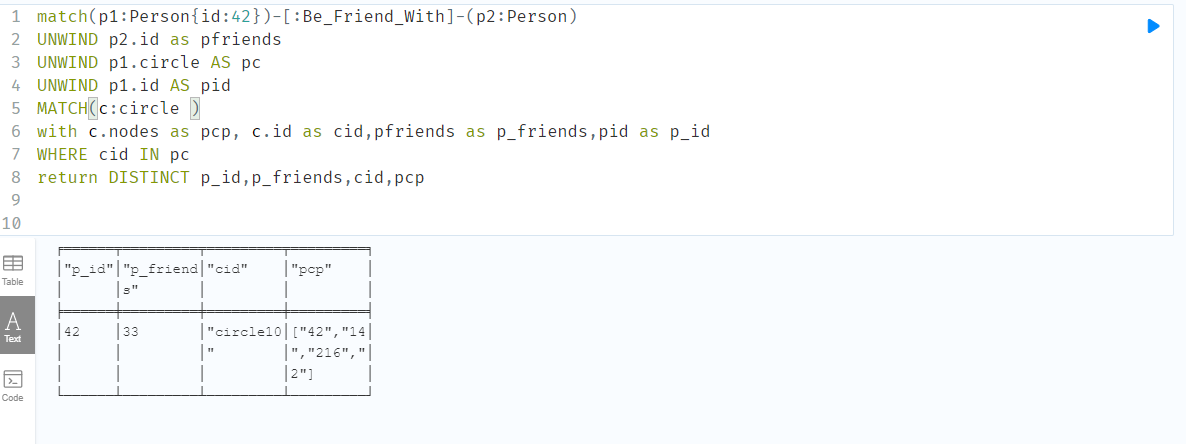
Id为1



Id为42



11. 检索某个Person的所有朋友集合和其所在的 circle的所有Person集合



12.任选三对Person，查找每一对Person间的最短关系链（即图模型的最短路）

Id为99



Id为295



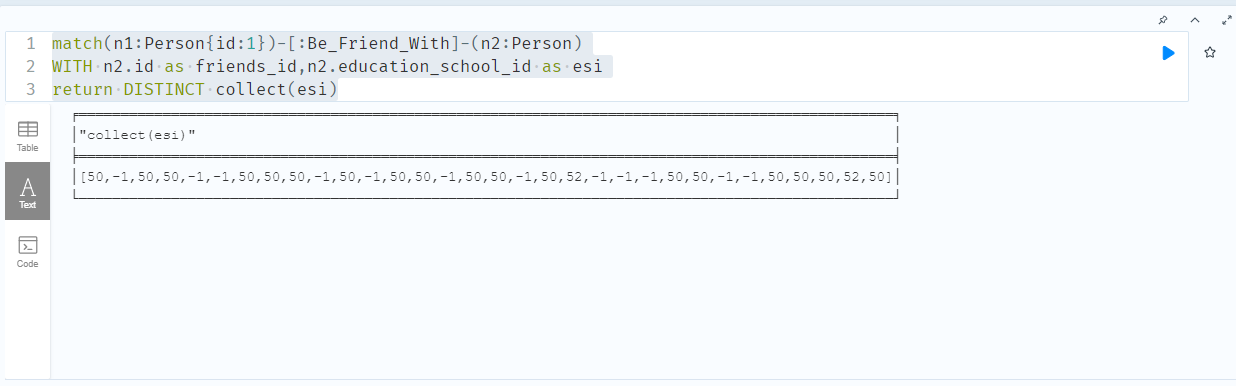
Id为236



1. 检索某个Person的二级和三级朋友集合

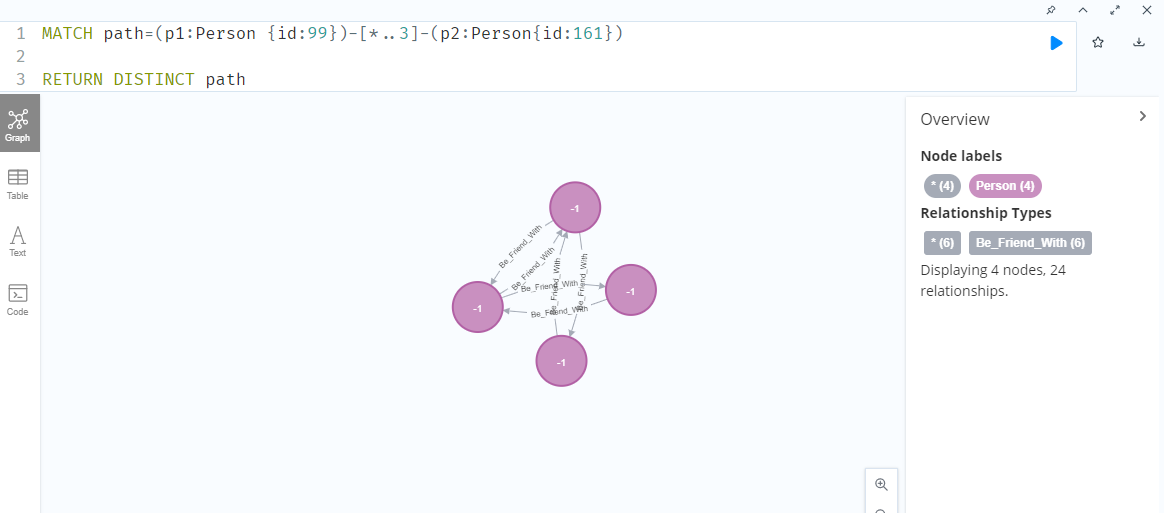


1. 获取某个Person的所有朋友education;school;id属性的list



1. 任选三对Person，查找每一对Person的关系路径中长度小于3的那些路径，检索出这些路径上年龄大于22的Person集合（因为电脑配置原因，小于10的路径查询速过慢，故改成3了）

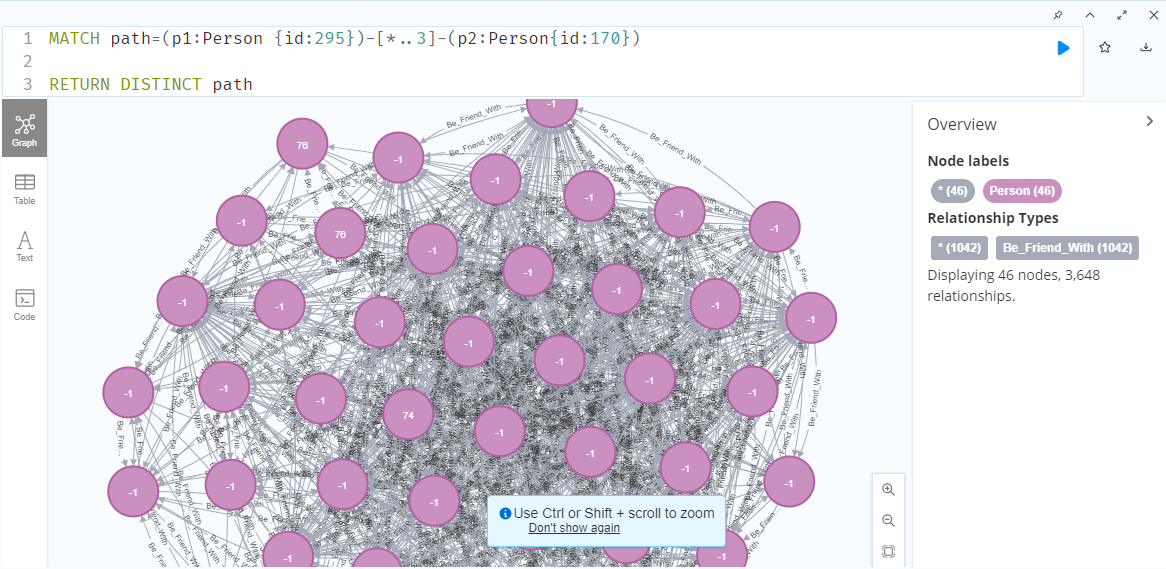
99-161间的路径



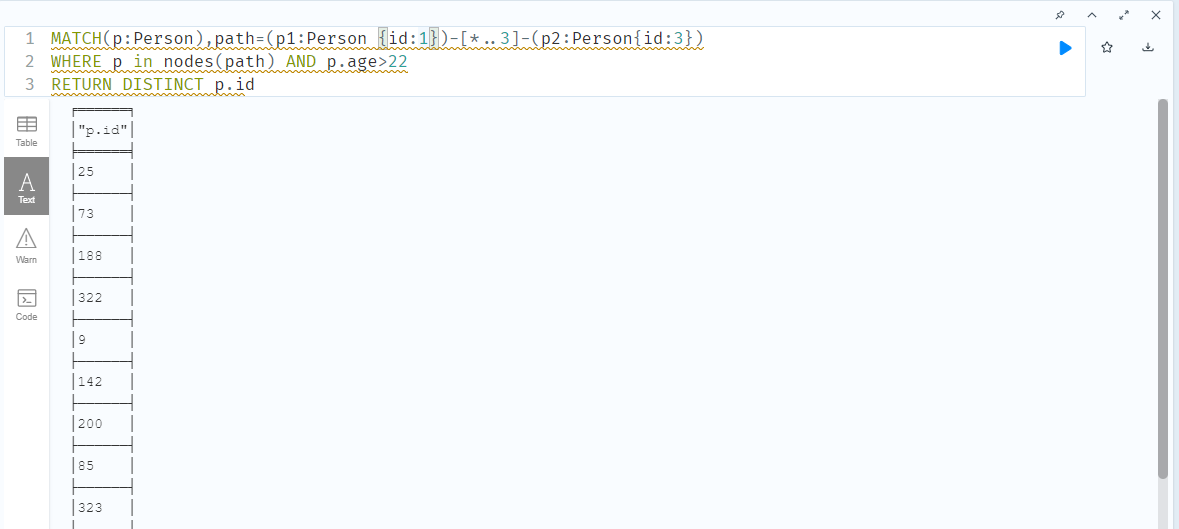
路径上age>22的集合



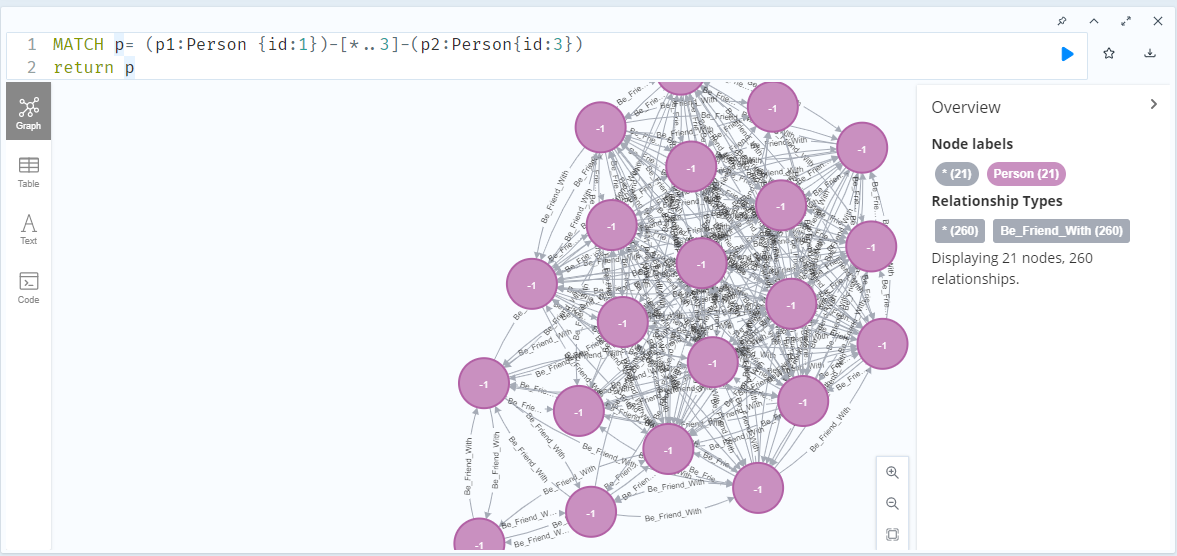
295-170间的路径



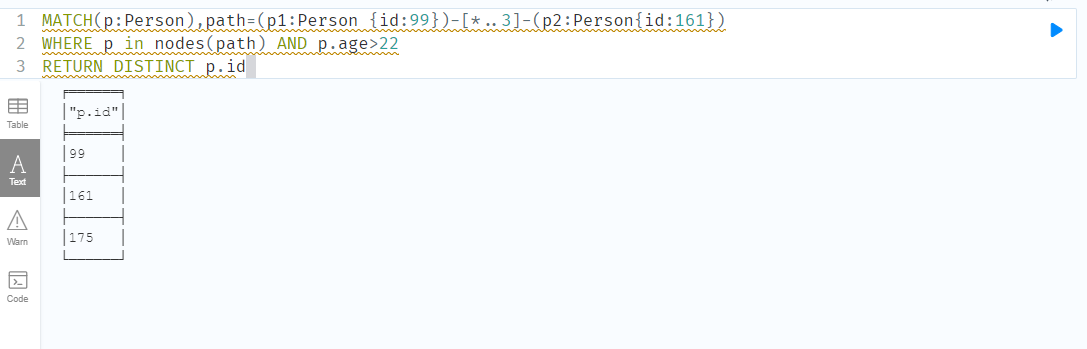
路径上age>22的集合



1-3间的路径



路径上age>22的集合



## 实验心得

数据的预处理比较麻烦，想处理成方便导入neo4j的形式需要注意很多细节。

Cypher语言的内容很多，学习时翻阅了手册上的许多语法和函数等内容，花了相当长的时间。