

2023-2024 学年 知识工程（双语）实验报告

任课教师：吴天星

院 系 人工智能学院

专 业 人工智能

姓 名 蒋雨初

任 务 Knowledge Querying

1 实验七

1.1 实验任务

针对下列自然语言，使用 SPARQL 查询语句返回结果:

- 1. List the creators (including paintings) of Guernica and Sunflowers, respectively.
- 2. List all the artists (including living places) who live in Spain or other places.
- 3. List all paintings, their names, and the corresponding techniques.

1.2 实验过程

1.2.1 Task 1

要分别列出 Guernica 和 Sunflowers 的作者及对应画作,只需要编写相应的 SPARQL 查询语句即可。Code 1展示了查询 Guernica 作者的查询语句，要分别查询 Sunflowers 的作者只需要把"Guernica"替换成"Sunflowers"即可。

```
1 PREFIX ex:<http://example.org/>
2 SELECT ?c ?p WHERE {
3     ?c ex:creatorOf ?p .
4     ?p rdfs:label "Guernica" .
5 }
```

Code 1: 查询 Guernica 创作者的 SPARQL 语句

结果展示在图 1。

Creator (including paintings) of Guernica	List all the artists (including living places) who live in Spain or other places.
?c = http://example.org/Picasso	?a = http://example.org/Picasso
?p = http://example.org/guernica	?p = "Spain"
Creator (including paintings) of Sunflowers	List all the artists (including living places) who live in Spain or other places.
?c = http://example.org/VanGogh	?a = http://example.org/VanGogh
?p = http://example.org/sunflowers	?p = null

图 1: 任务一的查询结果

图 2: 任务二的结果

1.2.2 Task2

要列出所有居住在西班牙或其他地方的艺术家 (包括居住地), 需要使用OPTIONAL以匹配没有所在地信息的艺术家, 如Code 2所示。其中, ?a a ex:Artist中的a是rdf:type的缩写。

```
1 PREFIX ex:<http://example.org/>
2 SELECT ?a ?p WHERE {
```

```

3      ?a a ex:Artist .
4  OPTIONAL {
5      ?a ex:homeAddress ?x .
6      ?x ex:country ?p .
7  }
8  }

```

Code 2: 查询艺术家及其所在地的 SPARQL 语句

结果展示在图 2。

1.2.3 Task 3

要列出所有画作、它们的名称以及相应的技法，只需要编写如Code 3SPARQL 语句即可。

```

1  PREFIX ex:<http://example.org/>
2  SELECT ?p ?n ?t WHERE {
3      ?p a ex:Painting ;
4      rdfs:label ?n ;
5      ex:technique ?t .
6  }

```

Code 3: 查询画作、它们的名称以及相应的技法的 SPARQL 语句

结果展示在图 3。

```

List all paintings, their names, and the corresponding techniques.
?p = http://example.org/guernica
?n = "Guernica"
?t = "oil on canvas"
?p = http://example.org/starryNight
?n = "Starry Night"
?t = "oil on canvas"
?p = http://example.org/sunflowers
?n = "Sunflowers"
?t = "oil on canvas"
?p = http://example.org/potatoEaters
?n = "The Potato Eaters"
?t = "oil on canvas"

```

图 3: 查询所有画作、名称以及相应技法的 SPARQL 语句

2 实验八

2.1 实验任务

1. 导入 contact-tracing-43.dump 文件到数据库 neo4j 中;
2. 查询名叫 Madison Odonnell 的人物节点, 并记录下该节点的 healthstatus、name、confirmedtimes 属性和属性值;
3. 将该人物节点及与其相连的关系删除, 并检查是否删除成功;
4. 重新创建该节点以及第 2 步记录下来的节点属性;
5. 重新创建关系: Madison Odonnell 的人物节点与名为 'Place nr40' 的 Place 节点间的关系, 不考虑关系属性;
6. Madison Odonnell 不幸被确诊为新冠 (healthstatus= 'sick'), 对图谱进行更新。

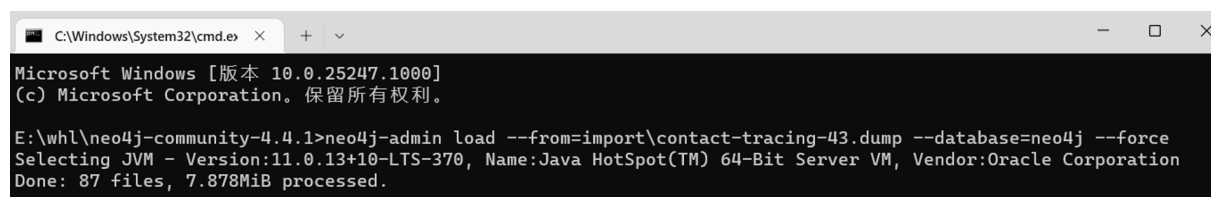
2.2 实验步骤

Step 1: 导入 contact-tracing-43.dump 文件到数据库 neo4j 中

把数据文件 movies-32.dump 和 contact-tracing-32.dump 移动到 neo4j/import 目录下, 然后打开 cmd 依次运行如下命令导入数据

```
neo4j stop
neo4j-admin load --from=import\contact-tracing-43.dump --database=neo4j
--force
```

运行结果如下图所示



Step 2: 查询名叫 Madison Odonnell 的人物节点, 并记录下该节点的 healthstatus、name、confirmedtimes 属性和属性值。先使用如下语句观察名叫 Madison Odonnell 的人物节点, 得到图 4。

```
MATCH (n:Person {name:'Madison Odonnell'})
RETURN p
```

再改写上述语句中的 RETURN p 为 RETURN n.healthstatus, n.name, n.confirmedtimes, 得到图 5。

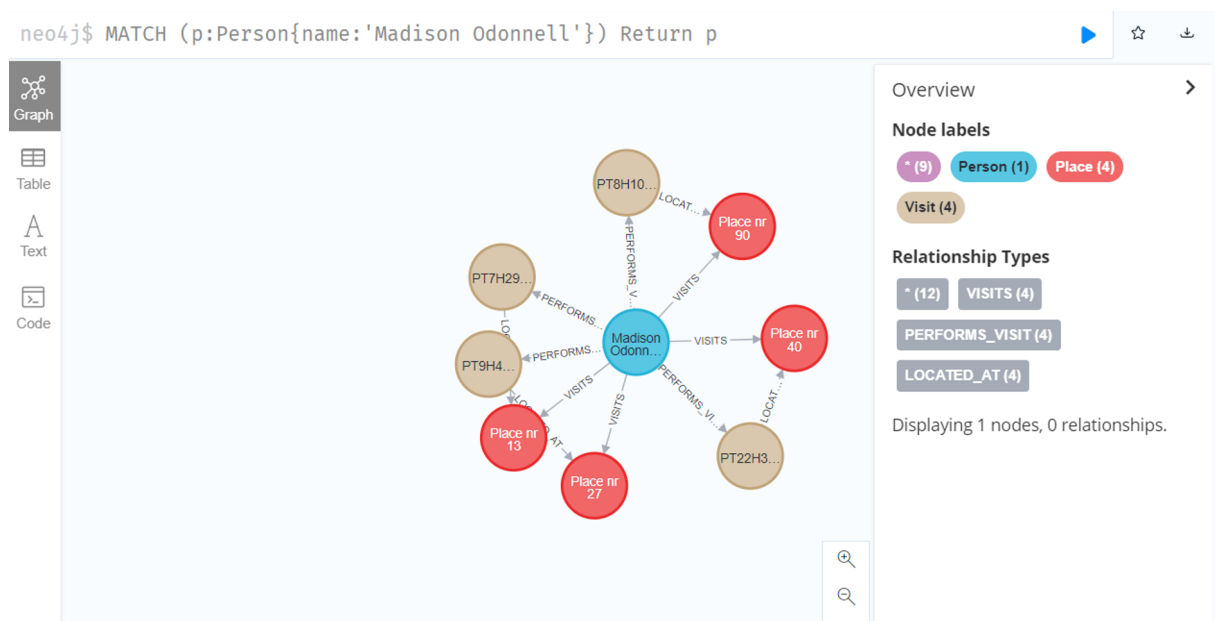


图 4: Madison Odonnell 的人物节点



图 5: Madison Odonnell 的 healthstatus、name、confirmedtimes 属性和属性值

Step 3: 将该人物节点及与其相连的关系删除，并检查是否删除成功

使用如下代码进行删除和重新查询，结果如图 6。

```
// delete all relations
MATCH (n:Person {name:'Madison Odonnell'})
DETACH DELETE n
// query result
MATCH (n:Person {name:'Madison Odonnell'})
RETURN n
```

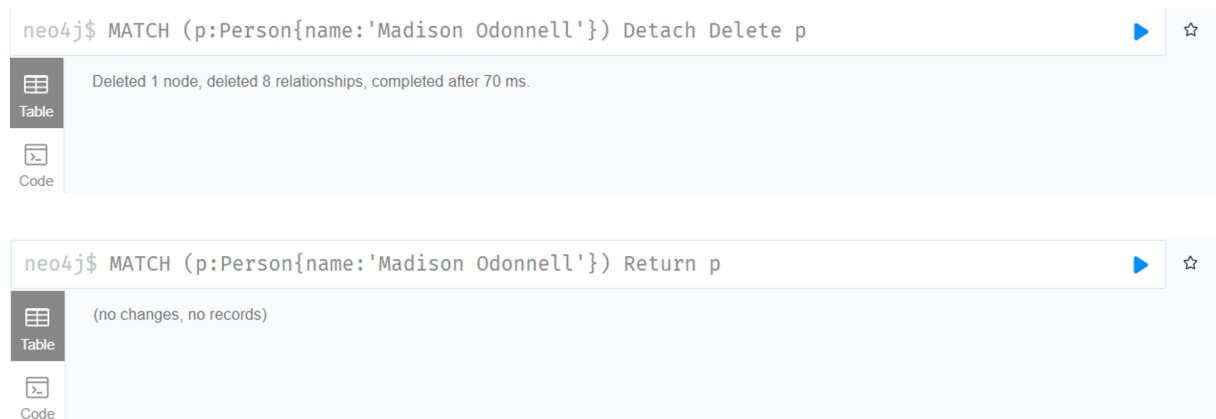


图 6: 删除 Madison Odonnell 的所有关系并进行检查。上方图为删除，下方图为检查是否删除成功。

Step 4: 重新创建该节点以及第 2 步记录下来的节点属性

创建节点需要使用 `CREATE` 语句，如下：

```
CREATE (
  n:Person {
    confirmedtime: "2020-04-25T23:09:38Z",
    healthstatus:"Healthy",
    name:"Madison Odonnell"
  }
)
RETURN n
```

从图 7 可见，重新创建了节点，并保持了原本的属性。

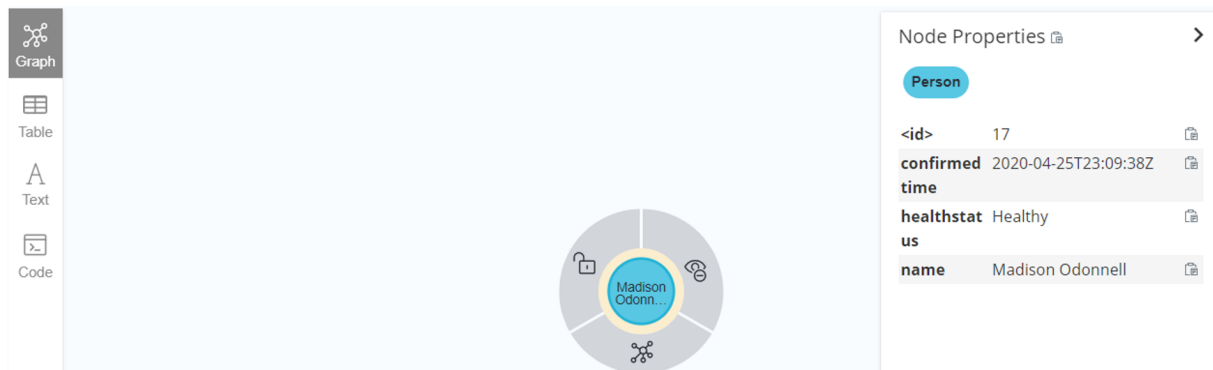


图 7: 重新创建出的新节点

Step 5: 重新创建关系

先使用**MATCH**匹配到两个相关的节点，再使用**CREATE**创建关系。

```
MATCH (p:Person {name:'Madison Odonnell'}), (p1:Place {name:'Place nr 40'})
CREATE (p)-[r:Visit]->(p1)
RETURN p, r, p1
```

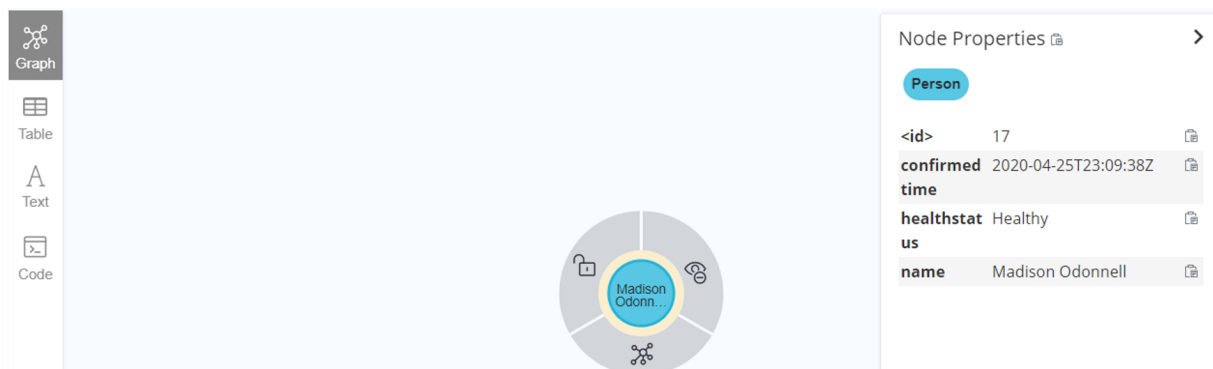


图 8: 创建出的新关系

Step 6: 更新图谱

要更新图谱，需要使用**SET**。

```
MATCH (p:Person {name:'Madison Odonnell'})
SET p.healthstatus='sick'
RETURN p
```

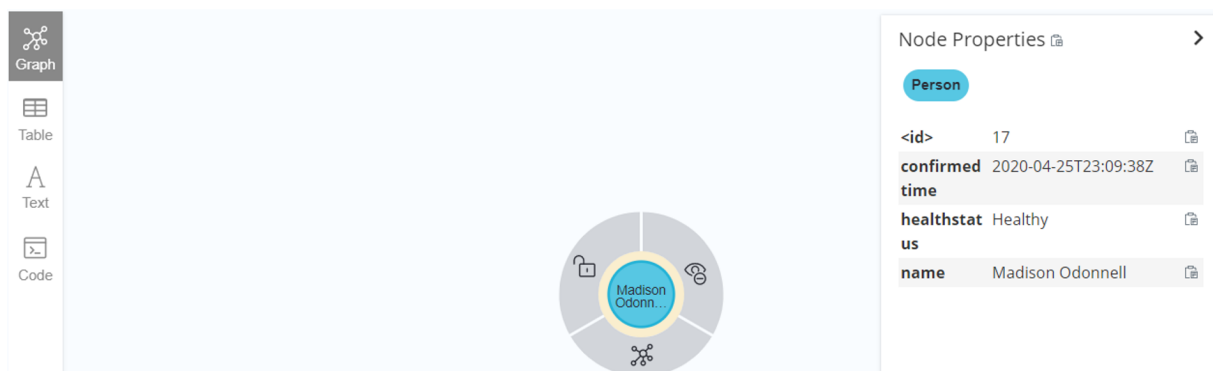


图 9: 更新节点的'healthstatus' 属性