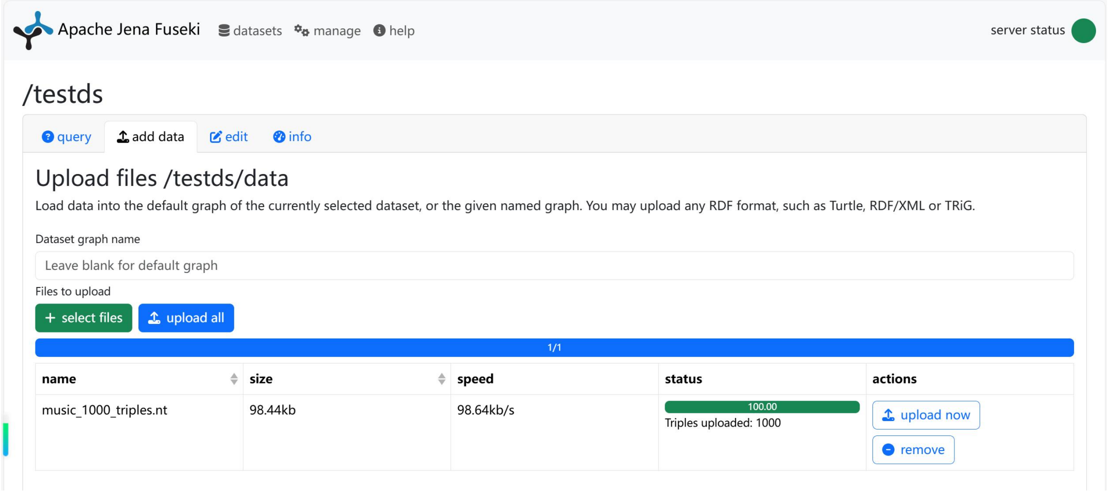
实验 3：Jena

1. 使用 Jena Fuseki 导入 music_1000_triples.nt 数据集，执行课程视频“实战-Jena.mp4”中所演示的所有 SPARQL 查询，并给出查询结果截图。

给出 music_1000_triples.nt 数据集导入成功截图：



/testds

Upload files /testds/data

Load data into the default graph of the currently selected dataset, or the given named graph. You may upload any RDF format, such as Turtle, RDF/XML or TRiG.

Dataset graph name

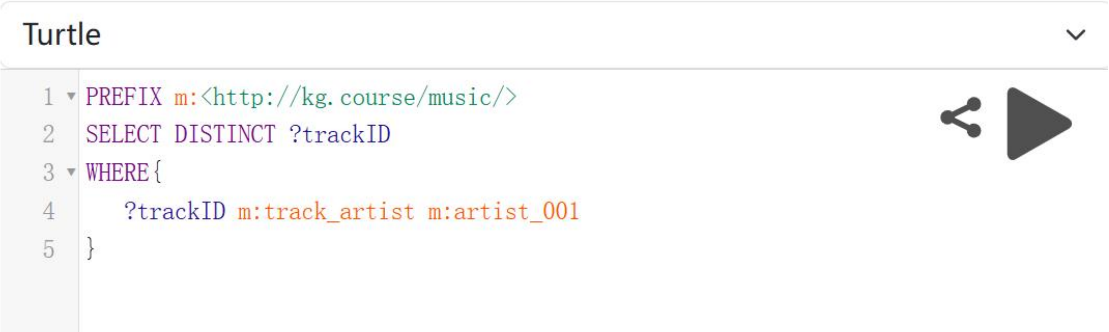
Leave blank for default graph

Files to upload

name	size	speed	status	actions
music_1000_triples.nt	98.44kb	98.64kb/s	100.00 Triples uploaded: 1000	upload now remove

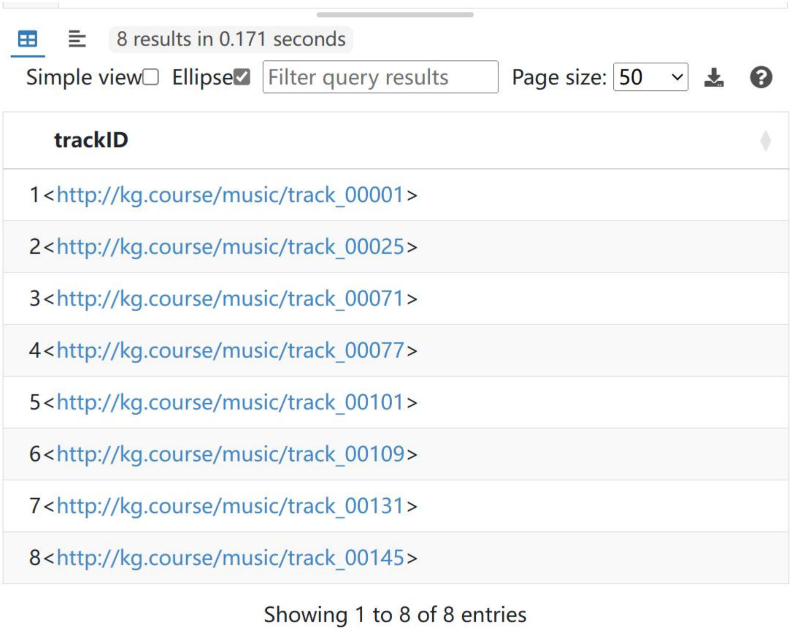
(1) 查询 1

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT DISTINCT ?trackID
WHERE{
    ?trackID m:track_artist m:artist_001
}
```

查询结果截图：



8 results in 0.171 seconds

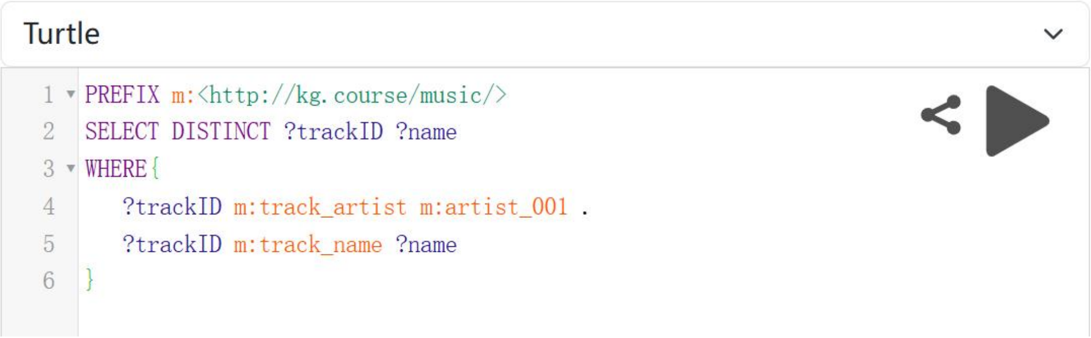
Simple view□ Ellipse☑ Filter query results Page size: 50

	trackID
1	<http://kg.course/music/track_00001>
2	<http://kg.course/music/track_00025>
3	<http://kg.course/music/track_00071>
4	<http://kg.course/music/track_00077>
5	<http://kg.course/music/track_00101>
6	<http://kg.course/music/track_00109>
7	<http://kg.course/music/track_00131>
8	<http://kg.course/music/track_00145>

Showing 1 to 8 of 8 entries

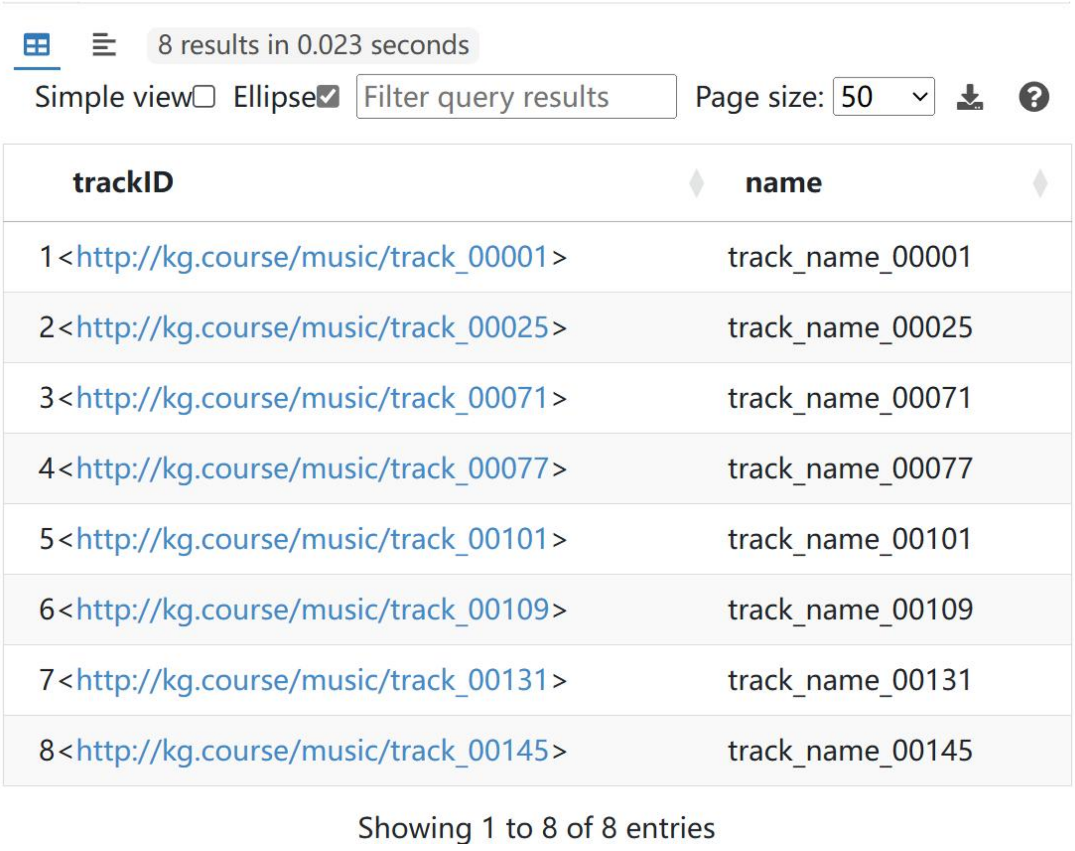
(2) 查询 2

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT DISTINCT ?trackID ?name
WHERE{
   ?trackID m:track_artist m:artist_001 .
   ?trackID m:track_name ?name
}
```

查询结果截图：

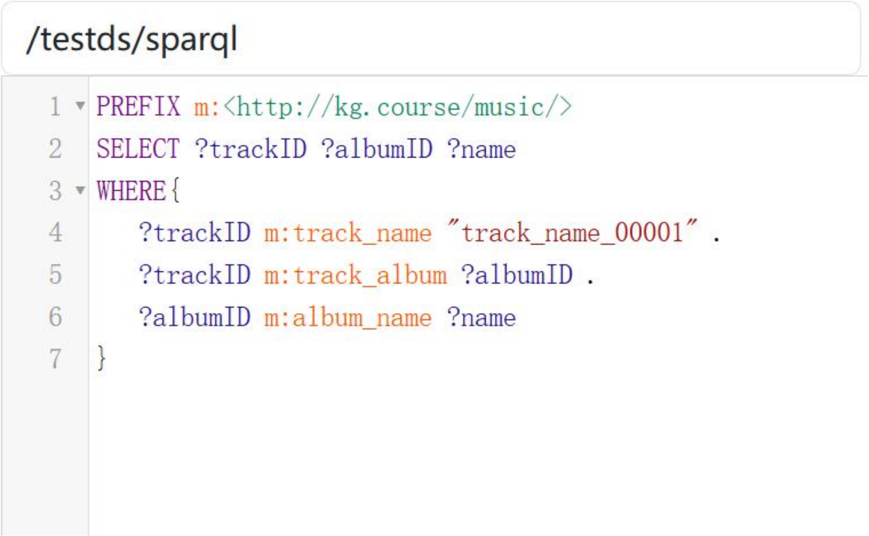


	trackID	name
1	<http://kg.course/music/track_00001>	track_name_00001
2	<http://kg.course/music/track_00025>	track_name_00025
3	<http://kg.course/music/track_00071>	track_name_00071
4	<http://kg.course/music/track_00077>	track_name_00077
5	<http://kg.course/music/track_00101>	track_name_00101
6	<http://kg.course/music/track_00109>	track_name_00109
7	<http://kg.course/music/track_00131>	track_name_00131
8	<http://kg.course/music/track_00145>	track_name_00145

Showing 1 to 8 of 8 entries

(3) 查询 3

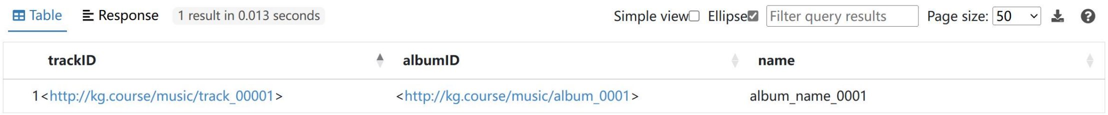
SPARQL 语句：



/testds/sparql

```
PREFIX m:<http://kg.course/music/>
SELECT ?trackID ?albumID ?name
WHERE{
    ?trackID m:track_name "track_name_00001" .
    ?trackID m:track_album ?albumID .
    ?albumID m:album_name ?name
}
```

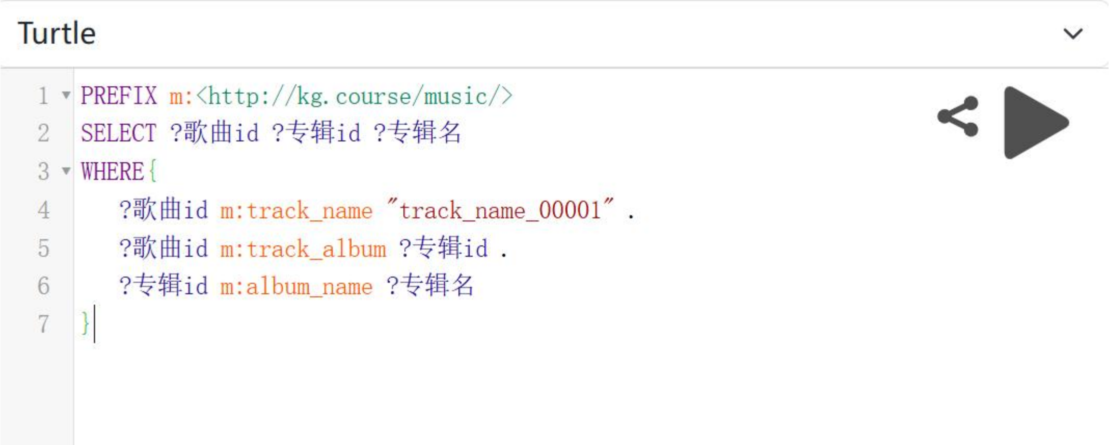
查询结果截图：



	trackID	albumID	name
1	<http://kg.course/music/track_00001>	<http://kg.course/music/album_0001>	album_name_0001

(4) 查询 4

SPARQL 语句：



Turtle

```
PREFIX m:<http://kg.course/music/>
SELECT ?歌曲id ?专辑id ?专辑名
WHERE{
   ?歌曲id m:track_name "track_name_00001" .
   ?歌曲id m:track_album ?专辑id .
   ?专辑id m:album_name ?专辑名
}
```

查询结果截图：

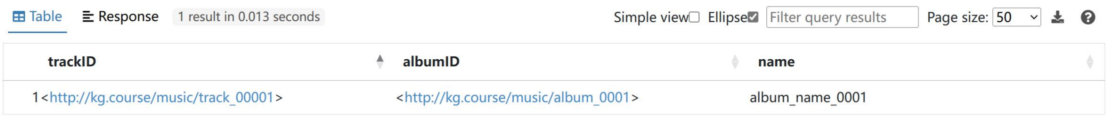
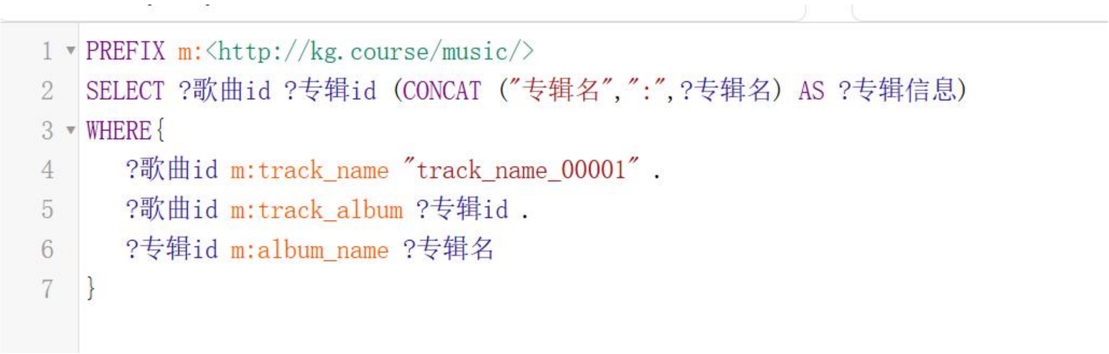


Table　Response　1 result in 0.013 seconds　Simple view□ Ellipse☑ Filter query results Page size: 50

	trackID	albumID	name
1	<http://kg.course/music/track_00001>	<http://kg.course/music/album_0001>	album_name_0001

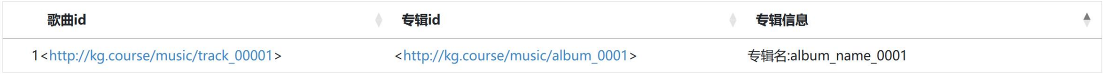
(5) 查询 5

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT ?歌曲id ?专辑id (CONCAT ("专辑名",":",?专辑名) AS ?专辑信息)
WHERE{
   ?歌曲id m:track_name "track_name_00001" .
   ?歌曲id m:track_album ?专辑id .
   ?专辑id m:album_name ?专辑名
}
```

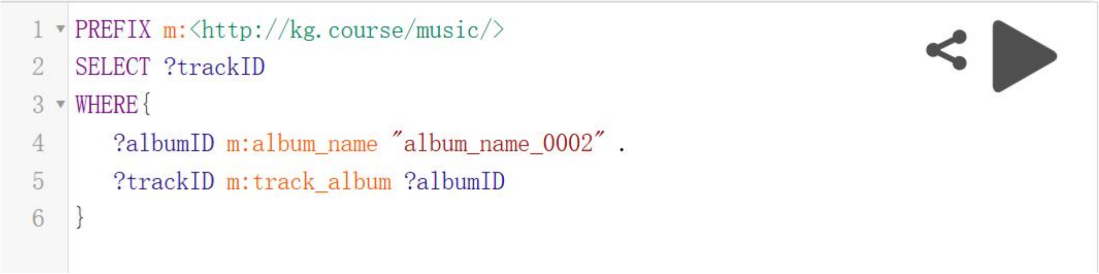
查询结果截图：



	歌曲id	专辑id	专辑信息
1	<http://kg.course/music/track_00001>	<http://kg.course/music/album_0001>	专辑名:album_name_0001

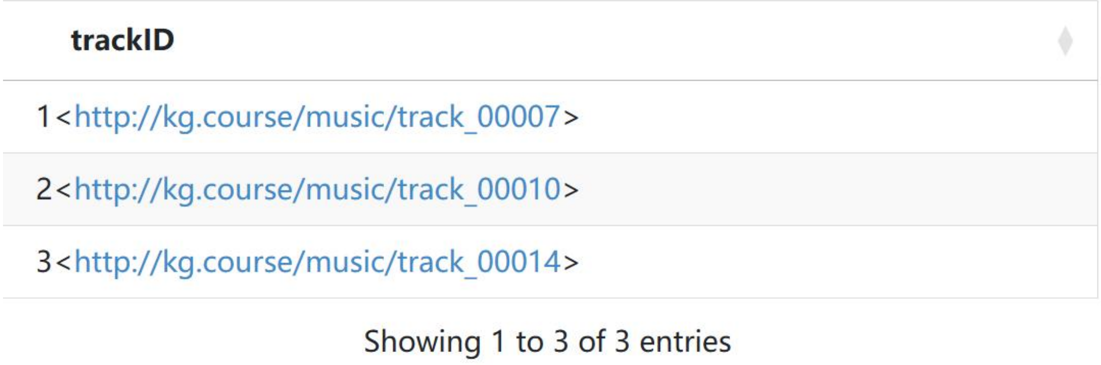
(6) 查询 6

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT ?trackID
WHERE{
   ?albumID m:album_name ″album_name_0002″ .
   ?trackID m:track_album ?albumID
}
```

查询结果截图：

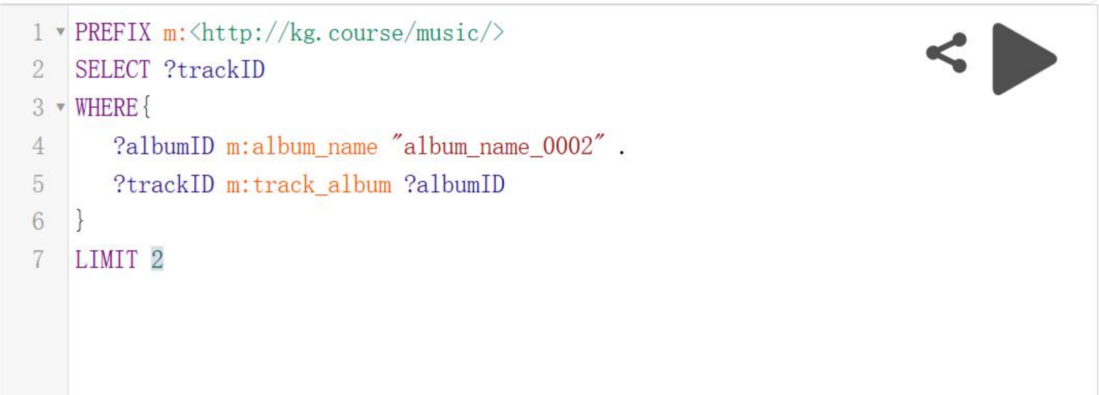


trackID
1<http://kg.course/music/track_00007>
2<http://kg.course/music/track_00010>
3<http://kg.course/music/track_00014>

Showing 1 to 3 of 3 entries

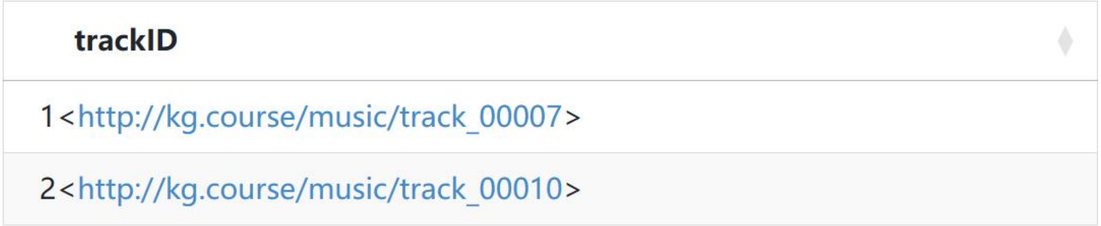
(7) 查询 7

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT ?trackID
WHERE{
   ?albumID m:album_name ″album_name_0002″ .
   ?trackID m:track_album ?albumID
}
LIMIT 2
```

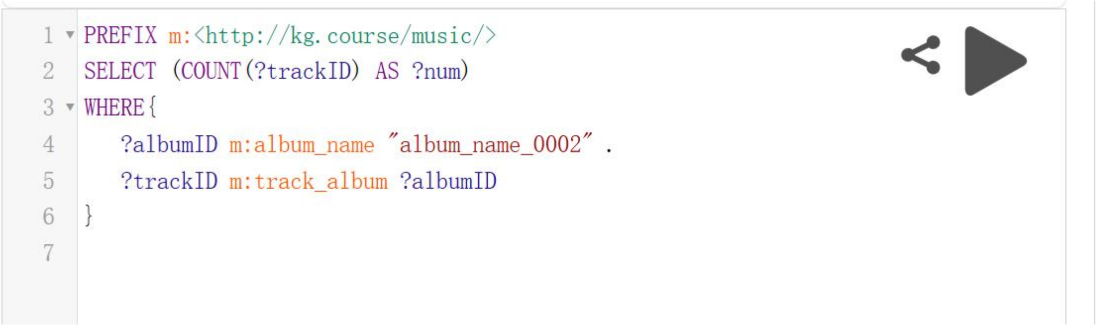
查询结果截图：



trackID
1<http://kg.course/music/track_00007>
2<http://kg.course/music/track_00010>

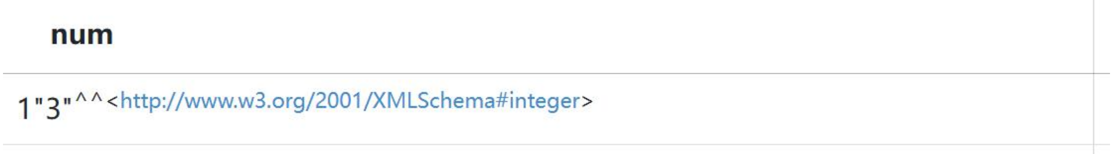
(8) 查询 8

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT (COUNT(?trackID) AS ?num)
WHERE{
   ?albumID m:album_name "album_name_0002" .
   ?trackID m:track_album ?albumID
}
```

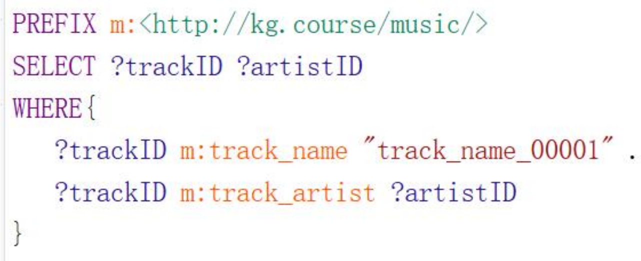
查询结果截图：



	num
1	"3"^^<http://www.w3.org/2001/XMLSchema#integer>

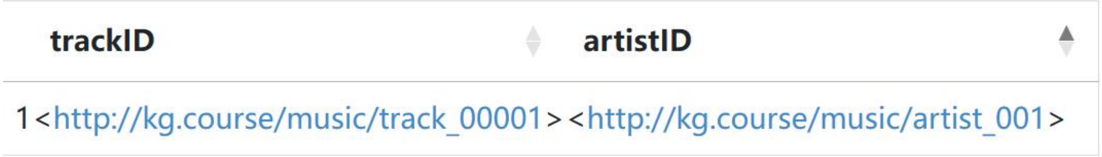
(9) 查询 9

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT ?trackID ?artistID
WHERE{
   ?trackID m:track_name "track_name_00001" .
   ?trackID m:track_artist ?artistID
}
```

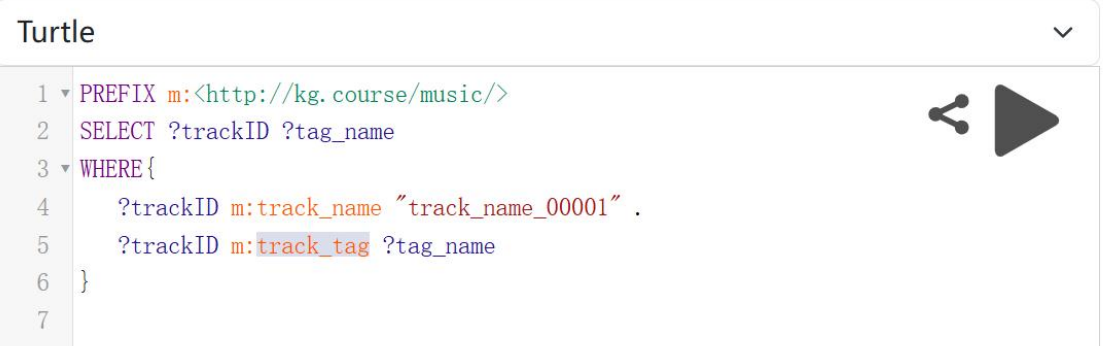
查询结果截图：



	trackID	artistID
1	<http://kg.course/music/track_00001>	<http://kg.course/music/artist_001>

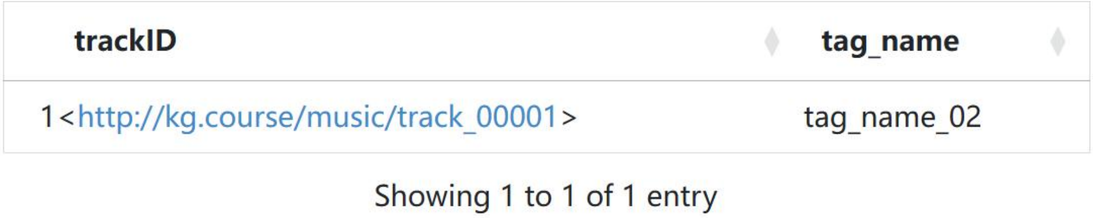
(10) 查询 10

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT ?trackID ?tag_name
WHERE{
   ?trackID m:track_name "track_name_00001" .
   ?trackID m:track_tag ?tag_name
}
```

查询结果截图：

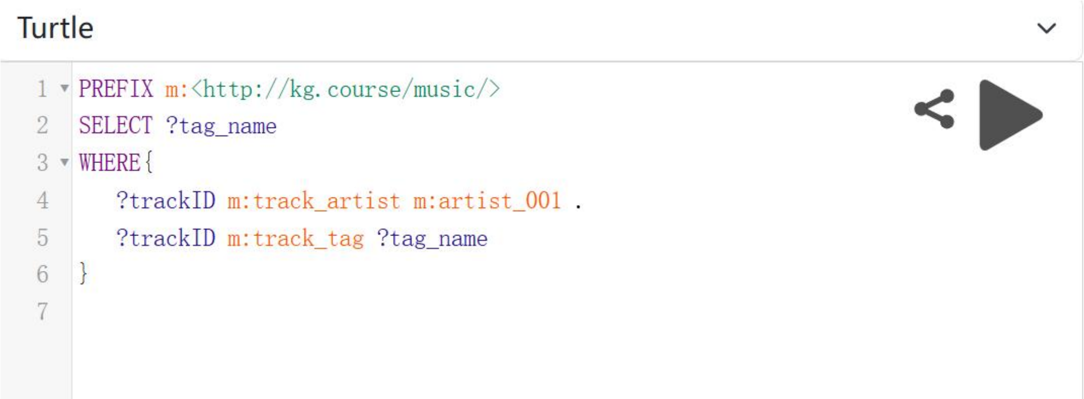


	trackID	tag_name
1	<http://kg.course/music/track_00001>	tag_name_02

Showing 1 to 1 of 1 entry

(11) 查询 11

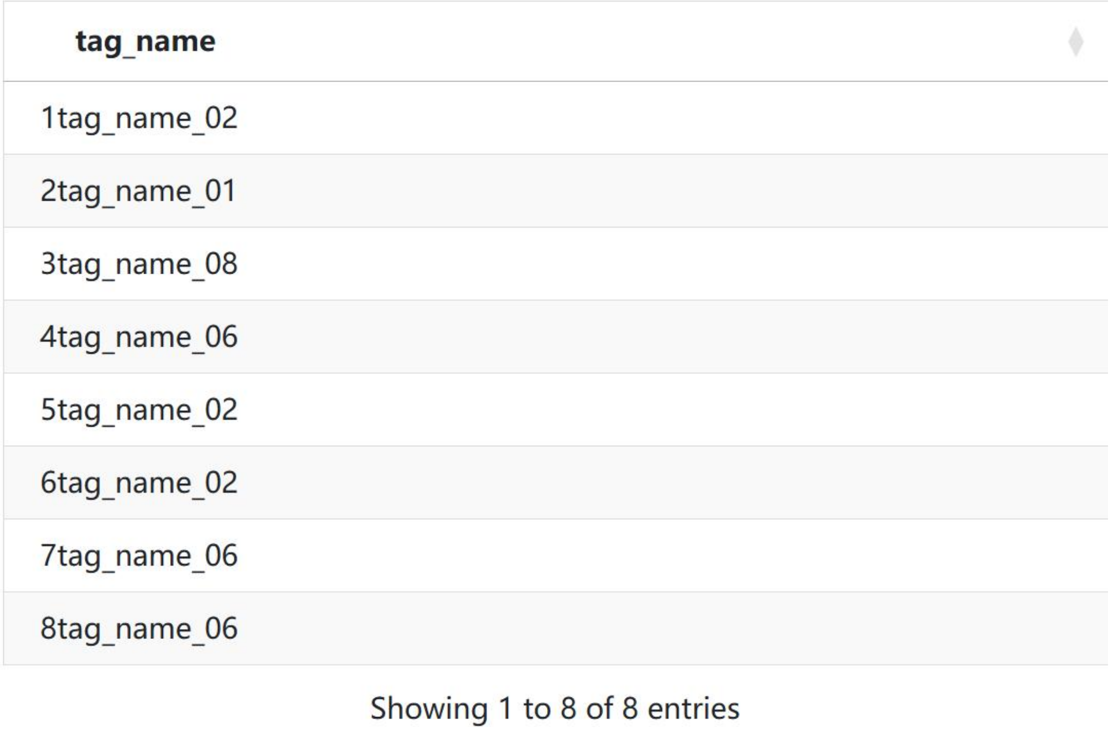
SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT ?tag_name
WHERE{
    ?trackID m:track_artist m:artist_001 .
    ?trackID m:track_tag ?tag_name
}

```

查询结果截图：

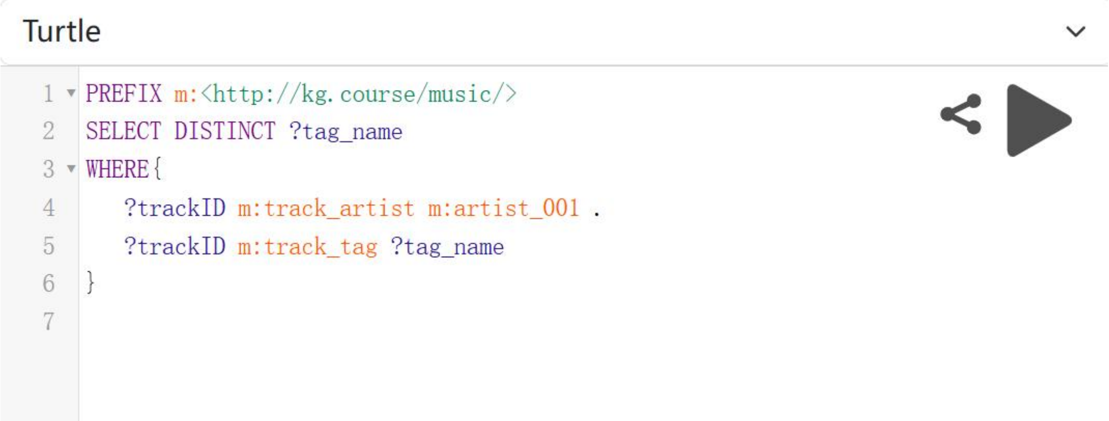


	tag_name
1	tag_name_02
2	tag_name_01
3	tag_name_08
4	tag_name_06
5	tag_name_02
6	tag_name_02
7	tag_name_06
8	tag_name_06

Showing 1 to 8 of 8 entries

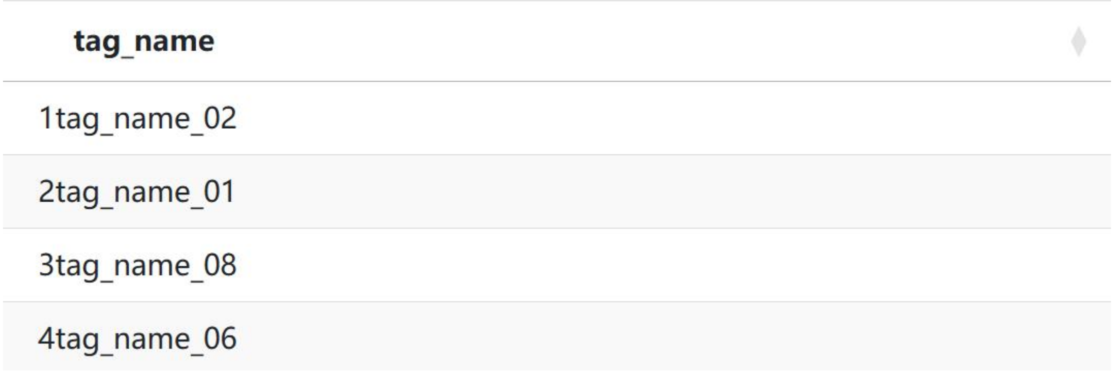
(12) 查询 12

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT DISTINCT ?tag_name
WHERE{
   ?trackID m:track_artist m:artist_001 .
   ?trackID m:track_tag ?tag_name
}
```

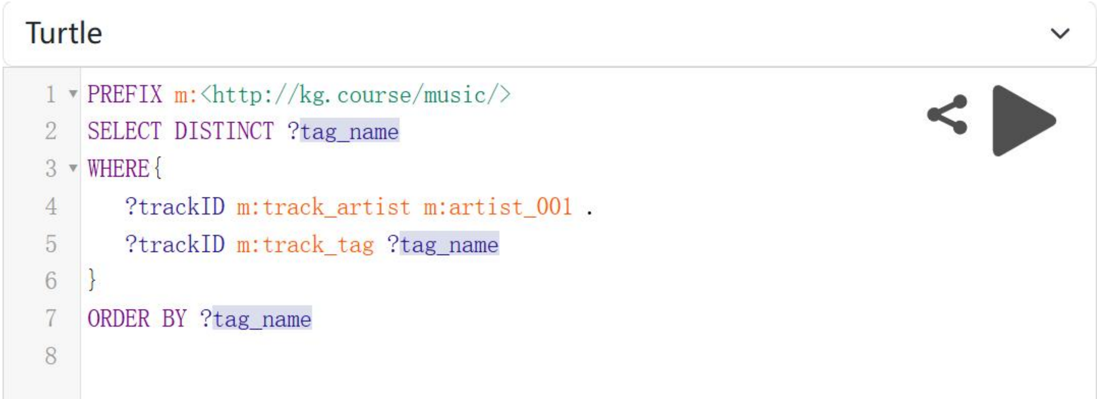
查询结果截图：



	tag_name
1	tag_name_02
2	tag_name_01
3	tag_name_08
4	tag_name_06

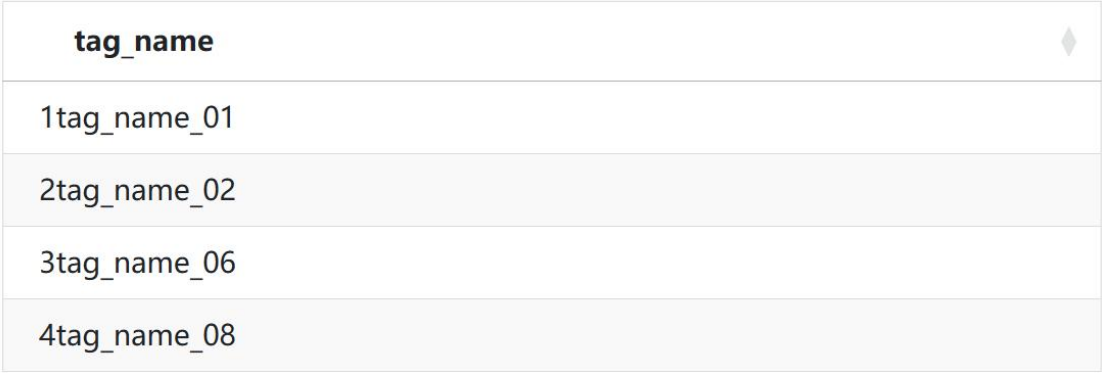
(13) 查询 13

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT DISTINCT ?tag_name
WHERE{
   ?trackID m:track_artist m:artist_001 .
   ?trackID m:track_tag ?tag_name
}
ORDER BY ?tag_name
```

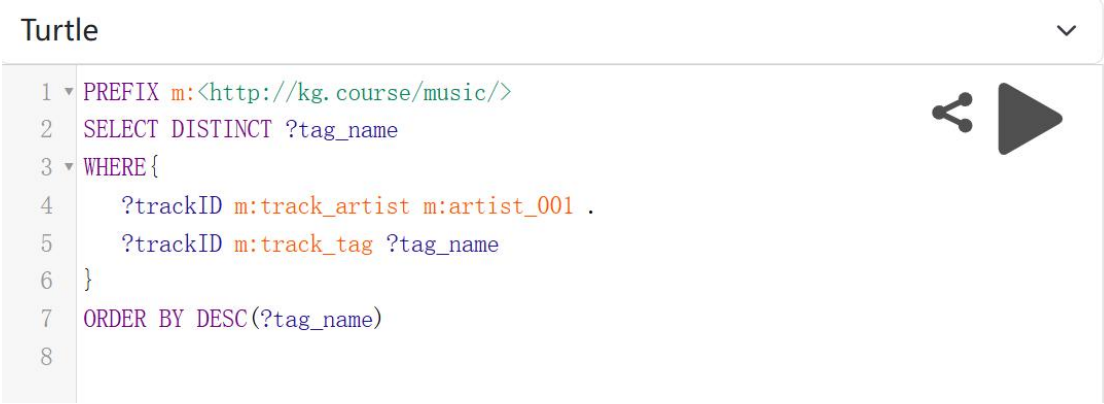
查询结果截图：



tag_name
1tag_name_01
2tag_name_02
3tag_name_06
4tag_name_08

(14) 查询 14

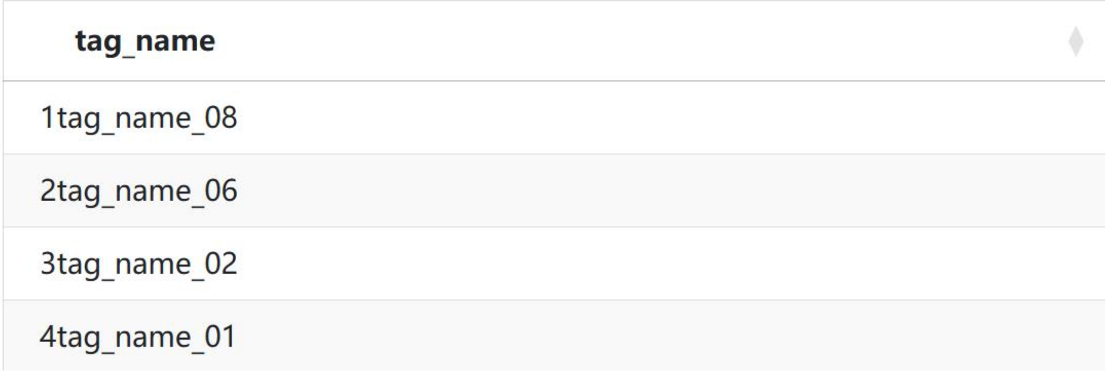
SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT DISTINCT ?tag_name
WHERE{
   ?trackID m:track_artist m:artist_001 .
   ?trackID m:track_tag ?tag_name
}
ORDER BY DESC(?tag_name)

```

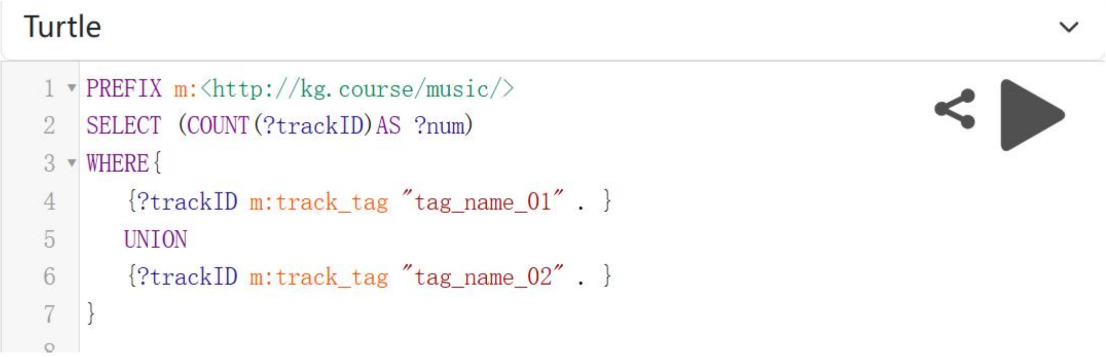
查询结果截图：



tag_name
1tag_name_08
2tag_name_06
3tag_name_02
4tag_name_01

(15) 查询 15

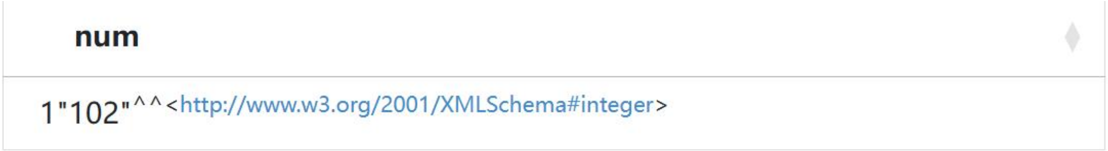
SPARQL 语句：



Turtle

```
PREFIX m:<http://kg.course/music/>
SELECT (COUNT(?trackID)AS ?num)
WHERE{
    {?trackID m:track_tag ″tag_name_01″ . }
    UNION
    {?trackID m:track_tag ″tag_name_02″ . }
}
```

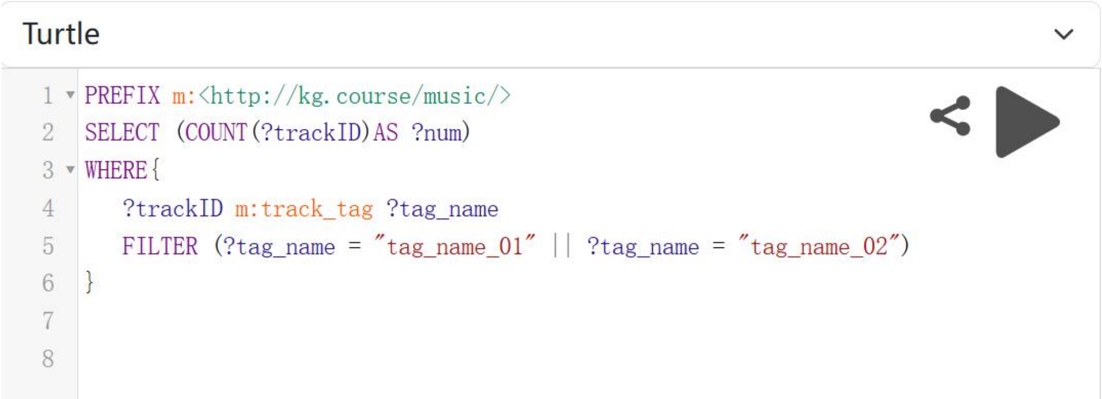
查询结果截图：



	num
1	"102"^^<http://www.w3.org/2001/XMLSchema#integer>

(16) 查询 16

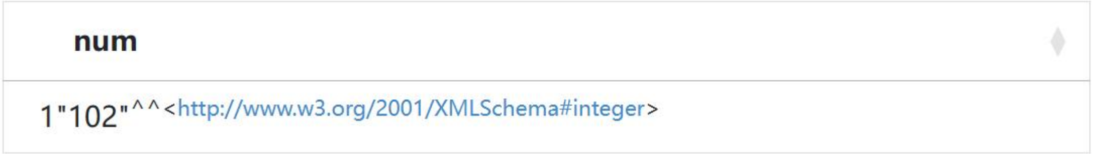
SPARQL 语句：



Turtle

```
PREFIX m:<http://kg.course/music/>
SELECT (COUNT(?trackID)AS ?num)
WHERE{
    ?trackID m:track_tag ?tag_name
    FILTER (?tag_name = ″tag_name_01″ || ?tag_name = ″tag_name_02″)
}
```

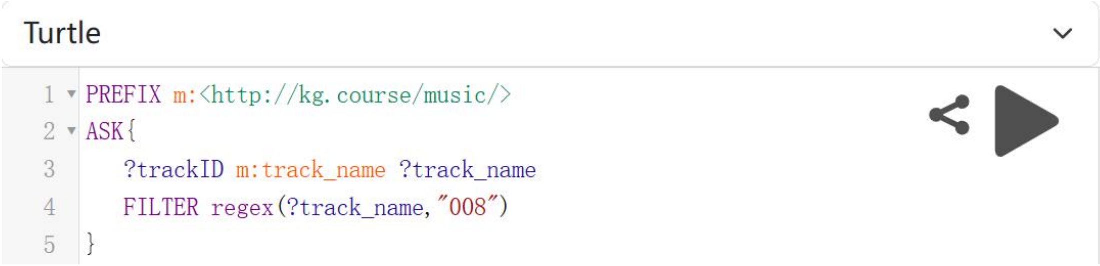
查询结果截图：



	num
1	"102"^^<http://www.w3.org/2001/XMLSchema#integer>

(17) 查询 17

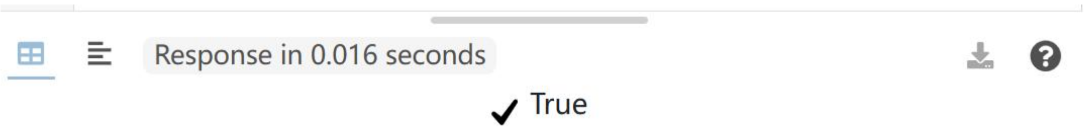
SPARQL 语句：



Turtle

```
PREFIX m:<http://kg.course/music/>
ASK{
    ?trackID m:track_name ?track_name
    FILTER regex(?track_name,″008″)
}
```

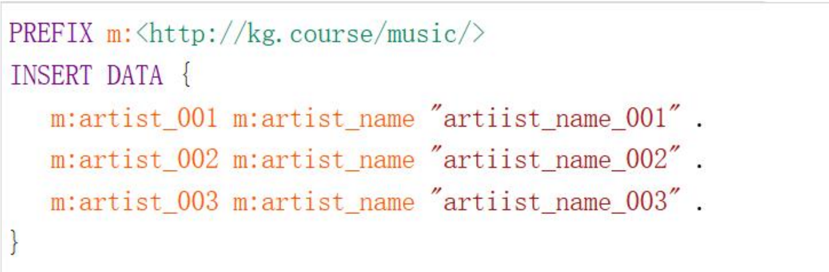
查询结果截图：



✓ True

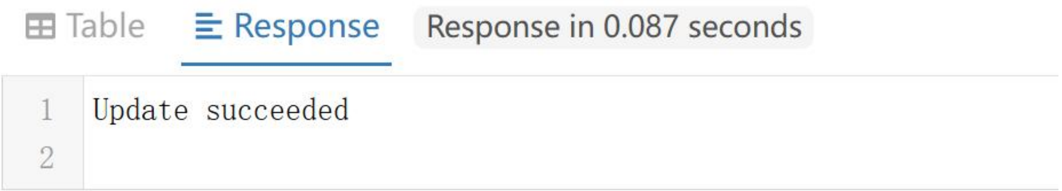
(18) 查询 18

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
INSERT DATA {
   m:artist_001 m:artist_name ”artiist_name_001” .
   m:artist_002 m:artist_name ”artiist_name_002” .
   m:artist_003 m:artist_name ”artiist_name_003” .
}
```

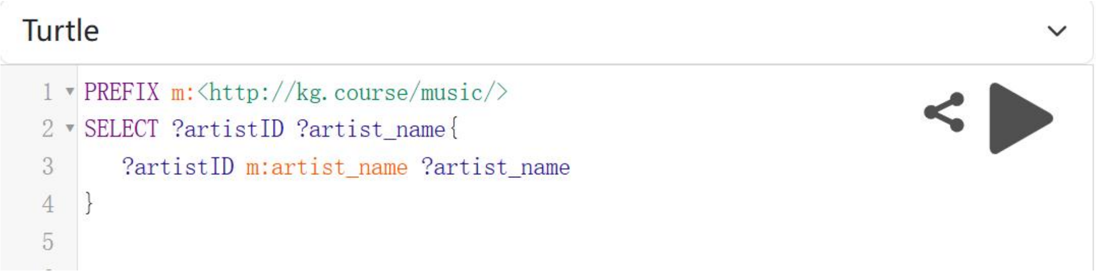
查询结果截图：



```
Update succeeded
```

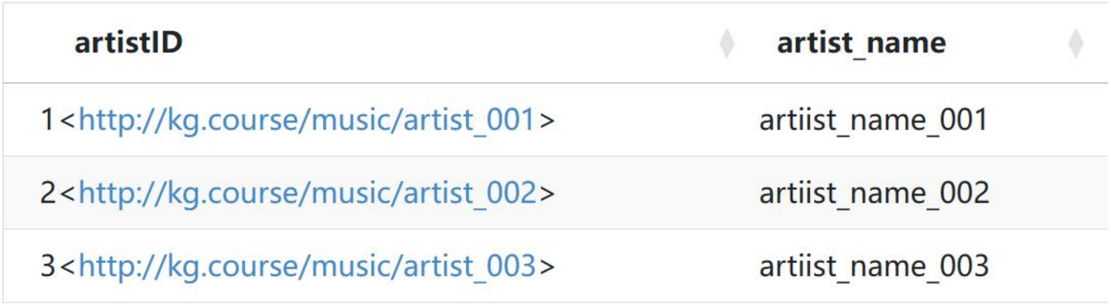
(19) 查询 19

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT ?artistID ?artist_name{
   ?artistID m:artist_name ?artist_name
}
```

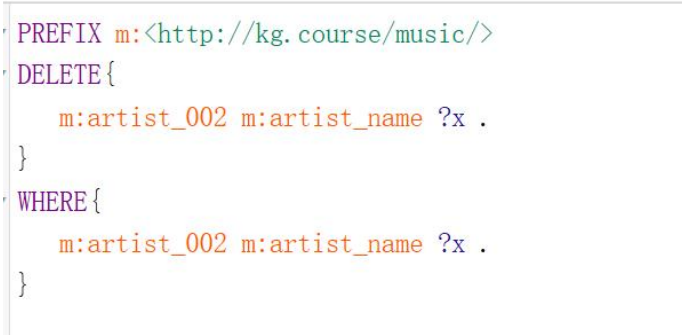
查询结果截图：



artistID	artist_name
1 <http://kg.course/music/artist_001>	artiist_name_001
2 <http://kg.course/music/artist_002>	artiist_name_002
3 <http://kg.course/music/artist_003>	artiist_name_003

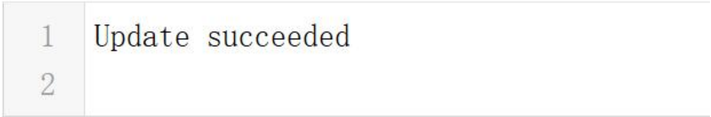
(20) 查询 20

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
DELETE{
   m:artist_002 m:artist_name ?x .
}
WHERE{
   m:artist_002 m:artist_name ?x .
}
```

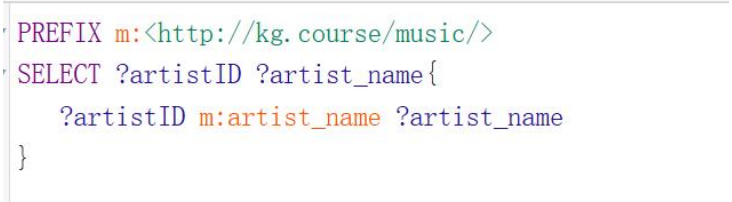
查询结果截图：



```
Update succeeded
```

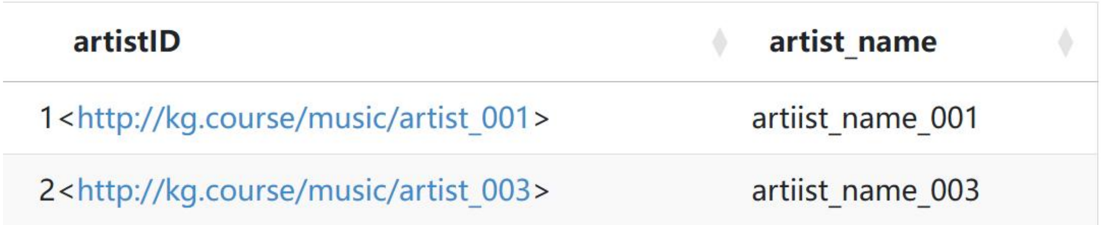
(21) 查询 21

SPARQL 语句：



```
PREFIX m:<http://kg.course/music/>
SELECT ?artistID ?artist_name{
   ?artistID m:artist_name ?artist_name
}
```

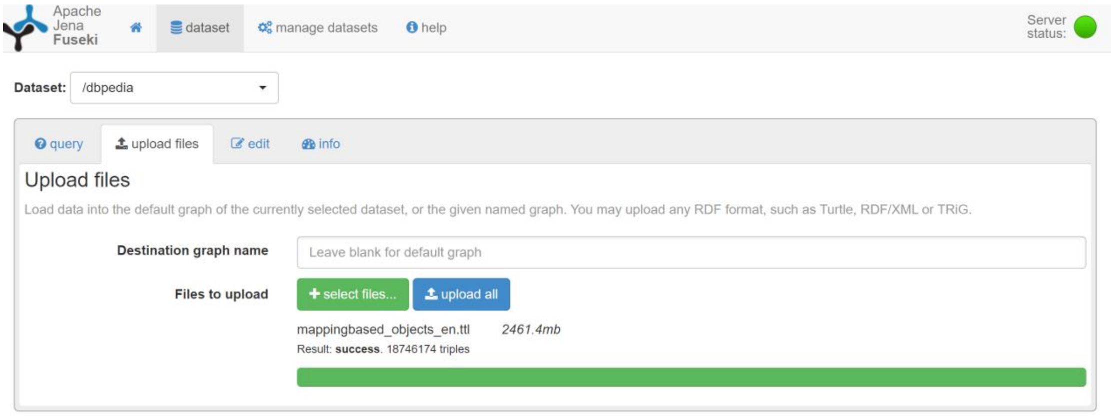
查询结果截图：



	artistID	artist_name
1	<http://kg.course/music/artist_001>	artiist_name_001
2	<http://kg.course/music/artist_003>	artiist_name_003

2. 使用 Jena Fuseki 导入 DBpedia 数据集（dbpedia-2016-10）的 mappingbased-objects_lang=en.ttl（bz2 压缩包需解压），编写 SPARQL 语句，执行以下查询问题，并给出查询结果截图。

mappingbased-objects_lang=en.ttl **数据集导入成功截图如下：**



Dataset: /dbpedia

Upload files

Load data into the default graph of the currently selected dataset, or the given named graph. You may upload any RDF format, such as Turtle, RDF/XML or TRiG.

Destination graph name: Leave blank for default graph

Files to upload

mappingbased_objects_en.ttl 2461.4mb

Result: **success**. 18746174 triples

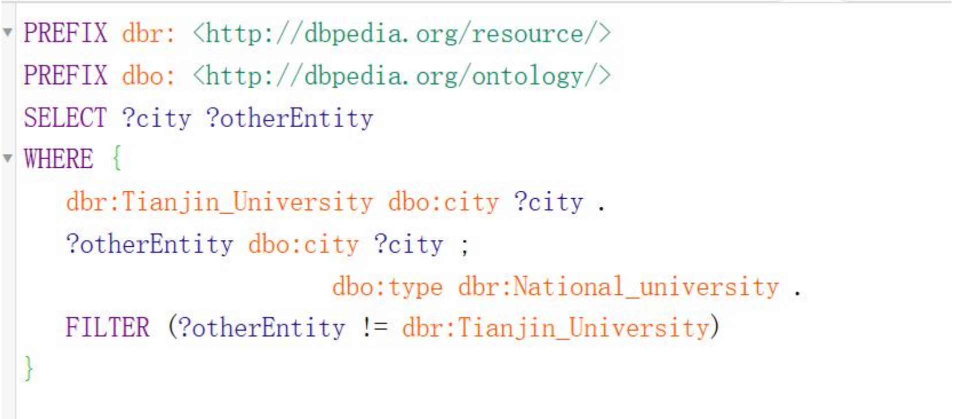
使用以下命名空间前缀:

PREFIX dbr: <http://dbpedia.org/resource/>

PREFIX dbo: <http://dbpedia.org/ontology/>

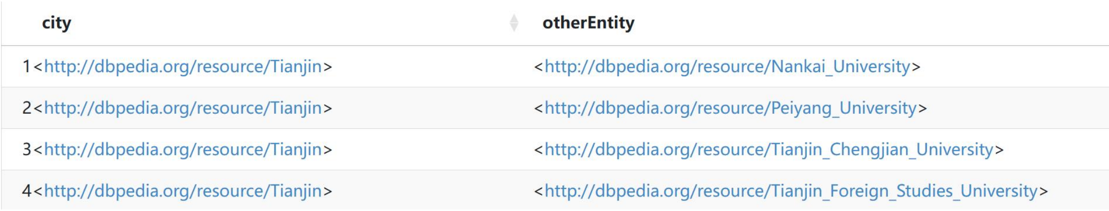
(1) 查询 dbr:Tianjin_University 所在城市（dbo:city）以及在同一城市的类型（dbo:type）为 dbr:National_university 的其他实体（dbr:Tianjin_University 自身不包括在结果中）。

SPARQL 语句：



```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?city ?otherEntity
WHERE {
   dbr:Tianjin_University dbo:city ?city .
   ?otherEntity dbo:city ?city ;
                    dbo:type dbr:National_university .
   FILTER (?otherEntity != dbr:Tianjin_University)
}
```

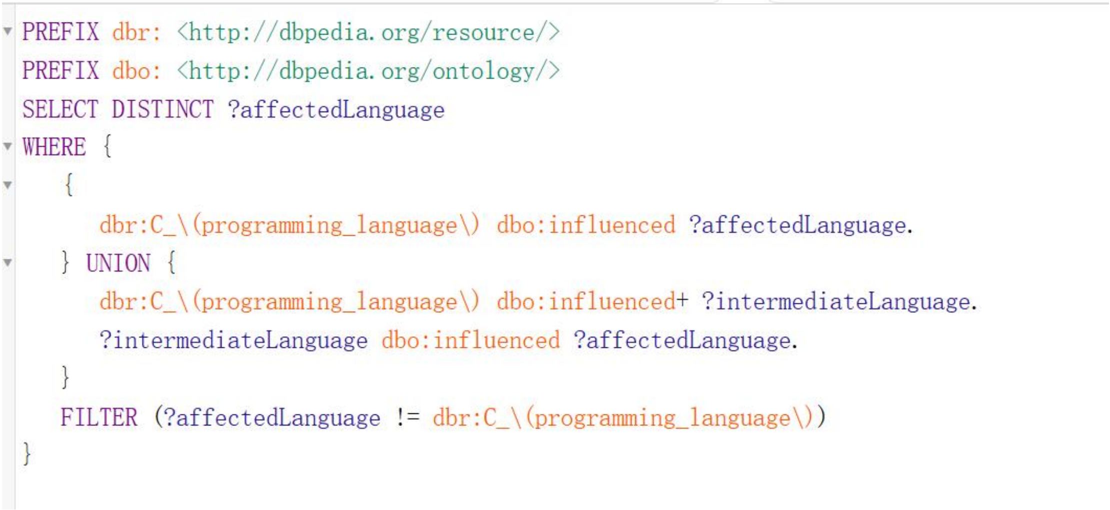
查询结果截图：



	city	otherEntity
1	<http://dbpedia.org/resource/Tianjin>	<http://dbpedia.org/resource/Nankai_University>
2	<http://dbpedia.org/resource/Tianjin>	<http://dbpedia.org/resource/Peiyang_University>
3	<http://dbpedia.org/resource/Tianjin>	<http://dbpedia.org/resource/Tianjin_Chengjian_University>
4	<http://dbpedia.org/resource/Tianjin>	<http://dbpedia.org/resource/Tianjin_Foreign_Studies_University>

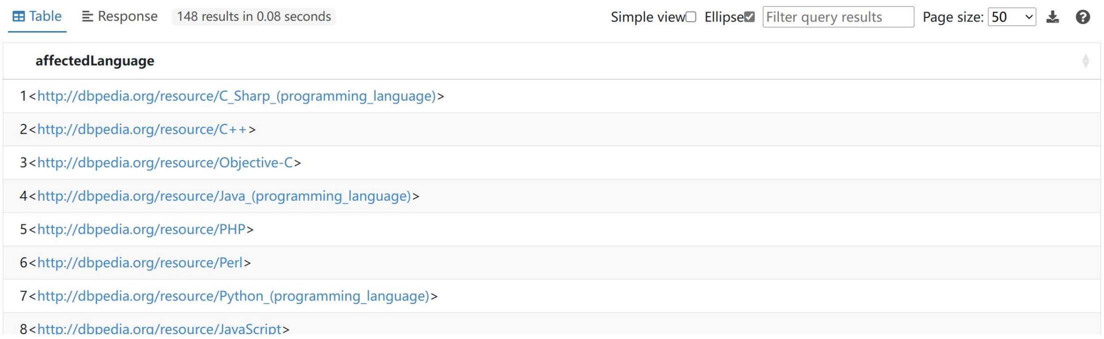
(2) 查询C程序设计语言dbr:C_\(programming_language\) 所直接和间接影响的所有其他程序设计语言（dbr:C_\(programming_language\)自身不包括在结果中）。

SPARQL 语句：



```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?affectedLanguage
WHERE {
   {
      dbr:C_\(programming_language\) dbo:influenced ?affectedLanguage.
   } UNION {
      dbr:C_\(programming_language\) dbo:influenced+ ?intermediateLanguage.
      ?intermediateLanguage dbo:influenced ?affectedLanguage.
   }
   FILTER (?affectedLanguage != dbr:C_\(programming_language\))
}
```

查询结果截图：
（只截图了前八条，共 148 条）



148 results in 0.08 seconds

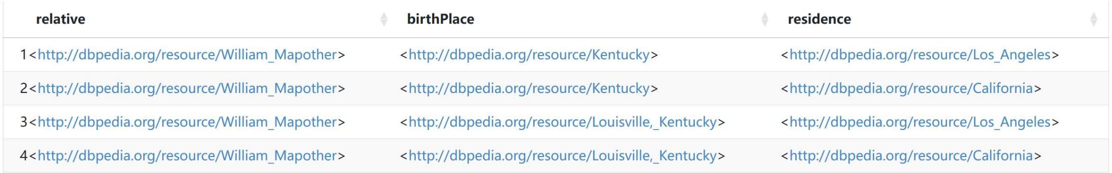
	affectedLanguage
1	<http://dbpedia.org/resource/C_Sharp_(programming_language)>
2	<http://dbpedia.org/resource/C++>
3	<http://dbpedia.org/resource/Objective-C>
4	<http://dbpedia.org/resource/Java_(programming_language)>
5	<http://dbpedia.org/resource/PHP>
6	<http://dbpedia.org/resource/Perl>
7	<http://dbpedia.org/resource/Python_(programming_language)>
8	<http://dbpedia.org/resource/JavaScript>

(3) 查询演员dbr:Tom_Cruise的邻居dbo:relative以及他们的出生地dbo:birthPlace和居住地dbo:residence（居住地如果有则输出，如果没用就空着）。

SPARQL 语句：

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?relative ?birthPlace ?residence
WHERE {
   dbr:Tom_Cruise dbo:relative ?relative.
   ?relative dbo:birthPlace ?birthPlace.
   OPTIONAL { ?relative dbo:residence ?residence }.
}
```

查询结果截图：



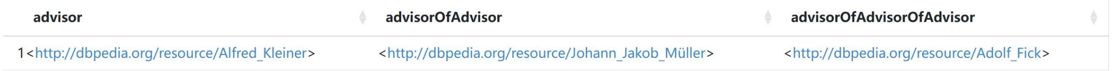
	relative	birthPlace	residence
1	<http://dbpedia.org/resource/William_Mapother>	<http://dbpedia.org/resource/Kentucky>	<http://dbpedia.org/resource/Los_Angeles>
2	<http://dbpedia.org/resource/William_Mapother>	<http://dbpedia.org/resource/Kentucky>	<http://dbpedia.org/resource/California>
3	<http://dbpedia.org/resource/William_Mapother>	<http://dbpedia.org/resource/Louisville,_Kentucky>	<http://dbpedia.org/resource/Los_Angeles>
4	<http://dbpedia.org/resource/William_Mapother>	<http://dbpedia.org/resource/Louisville,_Kentucky>	<http://dbpedia.org/resource/California>

(4) 查找爱因斯坦 dbr:Albert_Einstein 的导师（dbo:doctoralAdvisor）、导师的导师以及导师的导师的导师。

SPARQL 语句：

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?advisor ?advisorOfAdvisor ?advisorOfAdvisorOfAdvisor
WHERE {
   dbr:Albert_Einstein dbo:doctoralAdvisor ?advisor.
   OPTIONAL {
      ?advisor dbo:doctoralAdvisor ?advisorOfAdvisor.
      OPTIONAL {
         ?advisorOfAdvisor dbo:doctoralAdvisor ?advisorOfAdvisorOfAdvisor.
      }
   }
}
```

查询结果截图：



	advisor	advisorOfAdvisor	advisorOfAdvisorOfAdvisor
1	<http://dbpedia.org/resource/Alfred_Kleiner>	<http://dbpedia.org/resource/Johann_Jakob_Müller>	<http://dbpedia.org/resource/Adolf_Fick>