Atitit 维基数据查询语言sparql

目录

[1. 注释 # 1](#_Toc27210)

[1.1. 直接值与应用变量 1](#_Toc21532)

[2. Rdf 2](#_Toc16836)

[2.1. RDF 资源、属性和属性值 2](#_Toc422)

[2.2. RDF 陈述 3](#_Toc18659)

[2.3. 资源、属性和属性值的组合可形成一个陈述（被称为陈述的主体、谓语和客体）。 3](#_Toc14899)

[2.4. 语法与范例 3](#_Toc15902)

[3. 含义 4](#_Toc25590)

[4. 常见查询 4](#_Toc29200)

[4.1. 查询所有属性与属性值 获取单挑记录 4](#_Toc19344)

[4.2. 查询一个属性条件 5](#_Toc27544)

[4.3. 查询多个属性条件 wehre p1=1 and p2=2 5](#_Toc7703)

[4.4. Where语句的查询 FILTER 函数 6](#_Toc17989)

[5. 与sql的重要区别 6](#_Toc26730)

[5.1. rdf的定义对应关系 6](#_Toc25821)

[5.2. Rdf往往一个图集很少需要from，sql是多表格的 6](#_Toc28179)

[5.3. 显示列的制定，sparql是在where子句里面，sql是在select子句 6](#_Toc8497)

[5.4. And条件，sparql用分号连接，sql用and 6](#_Toc31813)

[6. Ref 6](#_Toc11182)

# 注释 #

SELECT ?country

WHERE

{

?country wdt:P463 wd:Q458.

#country #member of #European Union

}

## 直接值与应用变量

SELECT ?n WHERE {

?s rdf:type :Movie.

?s :movieTitle '英雄'.

?a :hasActedIn ?s.

?a :personName ?n

}

# Rdf

**RDF 使用 Web 标识符 (URIs) 来标识资源。**

**RDF 使用属性和属性值来描述资源。**

## **RDF 资源、属性和属性值**

RDF 使用 Web 标识符来标识事物，并通过属性和属性值来描述资源。

对资源、属性和属性值的解释：

* **资源**是可拥有 URI 的任何事物，比如 "http://www.w3school.com.cn/rdf"
* **属性**是拥有名称的资源，比如 "author" 或 "homepage"
* **属性值**是某个属性的值，比如 "David" 或 "http://www.w3school.com.cn" （请注意一个属性值可以是另外一个资源）

下面的 RDF 文档可描述资源 "http://www.w3school.com.cn/rdf"：

<?xml version="1.0"?>

<RDF>

<Description about="http://www.w3school.com.cn/RDF">

<author>David</author>

<homepage>http://www.w3school.com.cn</homepage>

</Description>

</RDF>

**提示：**上面是一个简化的例子。命名空间被忽略了。

## **RDF 陈述**

## 资源、属性和属性值的组合可形成一个**陈述**（被称为陈述的**主体**、**谓语**和**客体**）。

请看一些陈述的具体例子，来加深理解：

陈述："The author of http://www.w3school.com.cn/rdf is David."

* 陈述的主体是：http://www.w3school.com.cn/rdf
* 谓语是：author
* 客体是：David

## 语法与范例

SELECT \*

WHERE {

wd:Q9321259 wdt:P2048 ?属性1 .

}

查询对象的属性，属性可能是一个值，或者一个列表

SPARQL 的部分关键词：

* SELECT， 指定我们要查询的变量。在这里我们查询所有的变量，用 \* 代替。
* WHERE，指定我们要查询的图模式。含义上和 SQL 的 WHERE 没有区别。
* FROM，指定查询的 RDF 数据集。我们这里只有一个图，因此省去了 FROM 关键词。
* PREFIX，用于 IRI 的缩写。

在查询框里面，输入下面的 SPARQL 语句。

SELECT ?area

WHERE {

wd:Q46913 wdt:P150 ?area .}

上面代码要求返回变量 ?area，该变量必须满足主语 "山西省"（wd:Q46913）和谓语 "所包含的行政实体"（wdt:P150）。前缀 wd 表示这是维基数据的条目，而前缀 wdt 表示这是维基数据定义的谓语关系。

# 含义

RDF 的 含义. 大家都知道，关系型数据库 ...

前缀 wd 表示这是维基数据的条目，而前缀 wdt 表示这是维基数据定义的谓语关系。 点

## ?x 问号开头的都是变量，出现在 select 部分的变量会用于输出；

## where 条件中

的每一行都是一个三元组（SPO, Subject/Predicate/Object），以. 结束

# 常见查询

## 查询所有属性与属性值 获取单挑记录

SELECT \*

{

wd:Q9321259 ?属性 ?属性值1 .

}

SELECT \*

WHERE {

wd:Q9321259 ?属性 ?属性值1 .

}

查询某个属性

SELECT ?area

WHERE {

wd:Q46913 wdt:P150 ?area .}

SELECT \*

{

wd:Q46913 wdt:P150 ?绑定变量 .}

## 查询一个属性条件

SELECT ?资源主语 ?itemLabel

where{

?资源主语 wdt:P106 ‘555’ ;

## 查询多个属性条件 wehre p1=1 and p2=2

使用分号分割多个条件，主语可以沈略

SELECT ?资源主语 ?itemLabel

where{

?资源主语 wdt:P106 wd:Q488111 ;

wdt:P21 wd:Q6581072;

#sex:femal

wdt:P27 wd:Q17.

#control japan

SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO\_LANGUAGE],en". }

}

#country #职业 #pron actor

## Where语句的查询 FILTER 函数

使用 FILTER 正则匹配

SELECT DISTINCT ?pWHERE {

?p a rdf:Property.

?s ?p ?o.

FILTER (REGEX(str(?p), "date", "i")) # "i"代表大小写不敏感，匹配?p中出现date}

# 与sql的重要区别

## rdf的定义对应关系

资源（主语）》》表记录实体

属性（谓语）》》》字段

属性值（宾语）》》字段值

## Rdf往往一个图集很少需要from，sql是多表格的

## 显示列的制定，sparql是在where子句里面，sql是在select子句

## And条件，sparql用分号连接，sql用and

## SQ属性值常常是单一值，rdf可能是列表（List）

* 这些不能用 SQL 实现吗？

答：这里涉及到一个问题，属性值并不总是单一值（List），而 SQL 表是有这个假设的（比如 MySQL，PostgreSQL 是有 Array 类型的）。针对每一个多值属性都需要进行额外拆表，这对表的管理带来了巨大挑战。查

# 常见的在线rdf数据库

问题 2：RDF 数据如何管理？

RDF 本身是 schema-free 的，这的确给管理带来很大困难。

## 维基数据库

## schema.org 以及国内的 cnschema 可以作为一个起点，

尽量使用这些已经有良好定义的词典。我更倾向于在做数据的时候一类一类的数据对应分别的表，仅仅在最后面向使用时转换成 RDF 格式。

# 其他类似

问题 4：Neo4j 的 Cypher 怎么样？

也许很好，考虑到只能在 Neo4j 上使用，并且社区版的 Neo4j 只能跑在单机上，以及有无数号称速度超过 Neo4j 的图数据库已经出现了，个人不太想学。用 Gremlin 和 SPARQL 可以很容易地从某个数据库转到另外一个，但是 Cypher 就不要想了。另外，Neo4j 的数据组织是属性图的。

问题 5：MongoDB 和 ElasticSearch 呢？

你要是觉得写查询不累、构造查询不麻烦，其实都行的。抽象到像 SPARQL/Gremlin 这种级别的查询上还是需要一些工作的，本身未必很难，但是得做。

当然了，我个人其实是有明确倾向的，图查询还是用 SPARQL 吧。下一篇在讲讲常见的图数据和 ODBA 吧，AZA-AZA。

# Ref

SPARQL常用查询 - 简书.html

图数据库查询语言 Cypher、Gremlin 和 SPARQL\_数据库\_专注交流 - CSDN 博客.html