Atitit 知识图谱管理 谱存储选型 与查询

目录

[1. 知识图谱存储系统的选型。 1](#_Toc29891)

[1.1. 图数据库 neo4j 适合大规模数据 1](#_Toc8621)

[1.2. 关系数据库 小规模 2](#_Toc25304)

[2. 知识图谱查询语言 2](#_Toc3792)

[2.1. 。通常对于表达为RDF形式的知识图谱，可以使用SPARQL查询语言。 2](#_Toc9967)

[2.2. SQL语句就是比较好的选择。 2](#_Toc2858)

[3. SparQL](#_Toc5795) [3](#_Toc5795)

[4. 知识存储格式 3](#_Toc20965)

[4.1. RDF、RDFS和OWL 3](#_Toc13450)

[4.2. RDFa全称（RDF attribute 3](#_Toc1806)

[4.3. 微格式microformats（vCard（电子名片）和iCalendar hcard hCalendar 3](#_Toc14622)

[4.4. RSS 3](#_Toc6986)

[5. RDF序列化方法 3](#_Toc27677)

[5.1. RDF/XML，N-Triples，Turtle， 4](#_Toc12876)

[5.2. RDFa，JSON-LD等几种。 4](#_Toc6583)

[6. ref 5](#_Toc18879)

[6.1. Atitit 知识图谱 RDF、RDFS和OWL数据模型.docx 5](#_Toc2618)

# 知识图谱存储系统的选型。

## 图数据库 neo4j 适合大规模数据

知识图谱本质上在表达关联，天然地可以用图加以建模，因而很多人想到用图数据库对领域知识图谱加以存储。图数据库的确是知识图谱存储选型的重要选择，但是不是唯一选择。传统关系数据库，近几年充分发展的其他类型的NoSQL数据库在很多场景下也是合理选择。那么数据库的选择考虑的要素是什么呢？有两类重要的选型要素：图谱的规模以及操作复杂度。

从图谱的规模角度来看，百万、千万的节点和关系规模（以及以下规模）的图谱对于图数据库的需求并不强烈，图数据库的必要性在中等或者小规模知识图谱上体现并不充分。但是如果图谱规模在数亿节点规模以上，图数据库就十分必要了。

从操作复杂性来看，图谱上的操作越是复杂，图数据库的必要性越是明显。图谱上的全局计算（比如平均最短路径的计算），图谱上的复杂遍历，图谱上的复杂子图查询等等都涉及图上的多步遍历。图上的多步遍历操作如果是在关系数据库上实现需要多个联结（Join）操作。多个联结操作的优化一直以来是关系数据库的难题。图数据库系统实现时针对多步遍历做了大量优化，能够实现高效图遍历操作。

除了上述因素之外，还应该充分考虑系统的易用性、普及性与成熟度。总体而言图数据库还是发展中的技术，对于复杂图数据管理系统的优化也是只有少部分专业人员才能从事的工作。在数据库选型时需要充分考虑这些因素。我们实验室在实现CN-DBpedia（2000万实体、2.2亿关系）在线服务系统时先后采用了RelationalDB、Graph DB、MongoDB，最后出于综合考虑选用的是MongoDB，已经稳定运行了三年，累计提供10亿多次API服务。

## 关系数据库 小规模

# 知识图谱查询语言

## 。通常对于表达为RDF形式的知识图谱，可以使用SPARQL查询语言。

SPARQL语言针对RDF数据定义了大量的算子，对于推理操作有着很好支撑，因而能够适应领域中的复杂查询与复杂推理。从应用角度来看，也可以将知识图谱仅仅表达为无类型的三元组。对于这种轻量级的表示，关系数据库与传统NoSQL数据库也是较好选择。那么此时，

## SQL语句就是比较好的选择。

SQL十分成熟，语法简单，用户众多且有着几十年的成功应用基础。

很多领域图谱上的查询是相对简单的，以单步或者两到三步遍历居多。此时，SQL完全能够胜任。但是不排除有一些特定场景，特别是公共安全、风控管理等领域，通常需要进行复杂关联分析，需要较长路径的遍历，需要开展复杂子图挖掘，此时SQL的表达能力就显得相对较弱了。

---------------------

# SparQL

[编辑](https://baike.baidu.com/item/SparQL/javascript:;) [讨论](https://baike.baidu.com/planet/talk?lemmaId=10956405" \t "https://baike.baidu.com/item/SparQL/_blank)

本词条缺少****概述图****，补充相关内容使词条更完整，还能快速升级，赶紧来编辑吧！

本词条由[“科普中国”科学百科词条编写与应用工作项目](https://baike.baidu.com/science" \t "https://baike.baidu.com/item/SparQL/_blank) 审核 。

SPARQL [1]  (SPARQL Protocol and RDF Query Language)，是为[RDF](https://baike.baidu.com/item/RDF/5956080" \t "https://baike.baidu.com/item/SparQL/_blank)开发的一种查询语言和数据获取协议，它是为W3C所开发的RDF数据模型所定义，但是可以用于任何可以用RDF来表示的信息资源。SPARQL 协议和 RDF 查询语言（SPARQL）于2008年1月15日正式成为一项W3C推荐标准。SPARQL构建在以前的 RDF 查询语言（例如 rdfDB、RDQL 和 SeRQL）之上，拥有一些有价值的新特性

# 知识存储格式

## RDF、RDFS和OWL

## [RDFa](https://baike.baidu.com/item/RDFa" \t "https://baike.baidu.com/item/RDFa/_blank)全称（RDF attribute

## 微格式microformats（vCard（电子名片）和iCalendar hcard hCalendar

## RSS

。RSS具有微格式的部分优点，但限制还是比较多的，比如有限的元数据（标题、描述、URL等），不能更好地描述语义，不太容易与已存在的工具结合等。用微格式可以来聚合外部Blog，Flickr，YouTube，MapQuest，甚至MySpace里的内容。

# RDF序列化方法

RDF的表示形式和类型有了，那我们如何创建RDF数据集，将其序列化（Serialization）呢？换句话说，就是我们怎么存储和传输RDF数据。目前，RDF序列化的方式主要有：

## RDF/XML，N-Triples，Turtle，

## RDFa，JSON-LD等几种。

RDF/XML，顾名思义，就是用XML的格式来表示RDF数据。之所以提出这个方法，是因为XML的技术比较成熟，有许多现成的工具来存储和解析XML。然而，对于RDF来说，XML的格式太冗长，也不便于阅读，通常我们不会使用这种方式来处理RDF数据。

N-Triples，即用多个三元组来表示RDF数据集，是最直观的表示方法。在文件中，每一行表示一个三元组，方便机器解析和处理。开放领域知识图谱DBpedia通常是用这种格式来发布数据的。

Turtle, 应该是使用得最多的一种RDF序列化方式了。它比RDF/XML紧凑，且可读性比N-Triples好。

RDFa, 即“The Resource Description Framework in Attributes”，是HTML5的一个扩展，在不改变任何显示效果的情况下，让网站构建者能够在页面中标记实体，像人物、地点、时间、评论等等。也就是说，将RDF数据嵌入到网页中，搜索引擎能够更好的解析非结构化页面，获取一些有用的结构化信息。读者可以去这个页面感受一下RDFa，其直观展示了普通用户看到的页面，浏览器看到的页面和搜索引擎解析出来的结构化信息。

JSON-LD，即“JSON for Linking Data”，用键值对的方式来存储RDF数据。感兴趣的读者可以参考此网站。

---------------------

# ref

## Atitit 知识图谱 RDF、RDFS和OWL数据模型.docx