

NobleProg



BlazeGraph Intro

Installation and Query

The World's Local Training Provider

*NobleProg® Limited 2017
All Rights Reserved*

Intro of BlazeGraph

- Graph Database: Save linked Data

BlazeGraph	RMDBs
Namespace	DataBase
Prefix	Table
Node	Row
Property	Column

- WikiData: high-performance in large data
- Support RDF/SPARQL API

RDF: Resource Description Framework

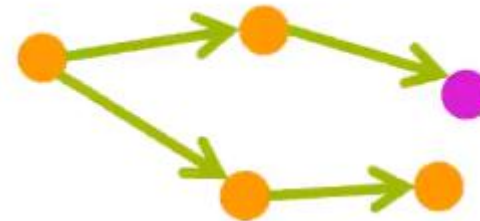
- Describe Web Resource / Data Format
- URI as name for ontology (global id)
 - <http://10.0.3.113:9999/blazegraph/#explore:kb:%3Chttp://nobleprog.schema.org/outline/12%3E>
- Describe the knowledge in a **Triple Tuple**.

- other way: eg.OWL

Presentation

- Knowledge Describe: **Triple Tuple** (Subject, Predicate, Object)
(Ontology, Relation, Ontology/Value)
(Vertex, Edge, Vertex)

TrainerA Name 'Anbo'
TrainerA Delivered Outline1



Formats

■ RDF/XML

```
<rdf:RDF
  xmlns:ex-schema=http://ex.org/schema#>
  <rdf:Description rdf:about="http://ex.org/ccf_adl">
    <ex-schema:speaker rdf:resource="http://ex.org/haofen"/>
    <ex-schema:theme rdf:resource="http://ex.org/KG"/>
  </rdf:Description>
</rdf:RDF>
```

– Turtle

```
@prefix ex: <http://ex.org/> .
@prefix ex-schema: <http://ex.org/schema#>
ex:ccf_adl
ex-schema:speaker ex:haofen;
ex-schema:theme ex:KG.
```

– N-Triples

```
<http://ex.org/ccf_adl>
  <http://ex.org/schema#speaker>
  <http://ex.org/haofen>.

<http://ex.org/ccf_adl>
  <http://ex.org/schema#theme>
  <http://ex.org/KG>.
```

JSON-LD
HTML5 Microdata

Storage — Blazgraph

- NanoSqaqlServer
blazegraph.jnl

com.bigdata.journal.Journal
bTree
(define how to do index/hashing/join)
- Embedded
eg. RDF4J

```
neostore.nodestore.db
neostore.nodestore.db.id
neostore.nodestore.db.labels
neostore.nodestore.db.labels.id
neostore.propertystore.db
neostore.propertystore.db.arrays
neostore.propertystore.db.arrays.id
neostore.propertystore.db.id
neostore.propertystore.db.index
neostore.propertystore.db.index.id
neostore.propertystore.db.index.keys
neostore.propertystore.db.index.keys.id
neostore.propertystore.db.strings
neostore.propertystore.db.strings.id
neostore.relationshipgroupstore.db
neostore.relationshipgroupstore.db.id
neostore.relationshipstore.db
neostore.relationshipstore.db.id
neostore.relationshiptypestore.db
neostore.relationshiptypestore.db.id
neostore.relationshiptypestore.db.names
neostore.relationshiptypestore.db.names.id
neostore.schemastore.db
neostore.schemastore.db.id
```

```
— GOSP.dat
— GOSP.idn
— GPOS.dat
— GPOS.idn
— GSP0.dat
— GSP0.idn
— journal.jrnl
— node2id.dat
— node2id.idn
— nodes.dat
— nodes.dat-jrnl
— OSP.dat
— OSPG.dat
— OSPG.idn
— OSP.idn
— POS.dat
— POSG.dat
— POSG.idn
— POS.idn
— prefix2id.dat
— prefix2id.idn
— prefixes.dat
— prefixes.dat-jrnl
— prefixIdx.dat
— prefixIdx.idn
— SPO.dat
— SPOG.dat
— SPOG.idn
— SPO.idn
— stats.opt
— tdb.lock
```

Run Service

- Nano Sparql Server

`java -server -Xmx4g -jar blazegraph.jar`

re-generate:

`https://github.com/blazegraph/database/blob/master/blazegraph-jar/src/main/java/com/bigdata/rdf/sail/webapp/StandaloneNanoSparqlServer.java`

- Vist the port using IPV4

eg: `http://172.17.0.1:9998/blazegraph/`

Parameters

- web.xml
- RWStore.properties
 - Djetty.port=9998
 - Djetty.overrideWebXml=/path/to/override.xml
 - Dbigdata.propertyFile=/etc/blazegraph/RWStore.properties

SPARQL: Grammar

- SPARQL:

```
SELECT ?a ?b ?t WHERE
{?a works_for ?u. ?b works_for ?u. ?a phd_from ?u. }
OPTIONAL {?a teaches ?t}
FILTER (regex(?u, "Saar"))
```

- SQL:

```
SELECT R1.A, R1.B, R2.T FROM
( SELECT P1.subject as A, P2.subject as B
FROM Triples P1, Triples P2, Triples P3
WHERE P1.predicate="works_for" AND P2.predicate="works_for"
AND P3.predicate="phd_from"
AND P1.object=P2.object AND P1.subject=P3.subject AND P1.object=P3.object
AND REGEXP_LIKE(P1.object, "Saar")
) R1 LEFT OUTER JOIN
( SELECT P4.subject as A, P4.object as T FROM Triples P4
WHERE P4.predicate="teaches") AS R2
) ON (R1.A=R2.A)
```

SPARQL: Query

- Q1
- Q2 count
- Q3 regex
- Q4 union
- Q5 optional

SPARQL: Insert

- Insert Language : Insert {?s ?p ?o}
- RDF file (ttl, n3)

SPARQL: Delete

- Delete Language : Delete {?s ?p ?o}
- Delete Based on Searching

SPARQL: Update

- Delete + Insert

Usage in Project

- Python

```
from pymantic import sparql
server = sparql.SPARQLServer('http://10.0.3.113:9999/blazegraph/sparql')

server.query(q)
server.update(q)
```



END THANKS