Herbert Wertheim College of Engineering
UNIVERSITY of FLORIDA

JENA DB

Group - 10

Abhishek Kumar
Harshvardhan Singh
Abhisek Mohanty
Suhas Tumkur Chandrashekhara

OUTLINE

- Introduction
- Data Model
- Query Language
- Implementation
- Features
- Applications

Introduction

- Open Source
- Java Framework
- Semantic Web
- Support for OWL

Data Model

RDF – Resource Description Framework

- Developed by the World Wide Web Consortium (W3C)
- Standard for representing vast amount of web
- Intended for applications processing web resources

History

- 1997 – Platform for Internet Content Selection PICS
- 1997 – Dublin Core and Meta Content Framework
- 1999 – 1st W3C Recommendation
- 2004 – RDF 1.0
- 2014 – RDF 1.1

RDF Data Model

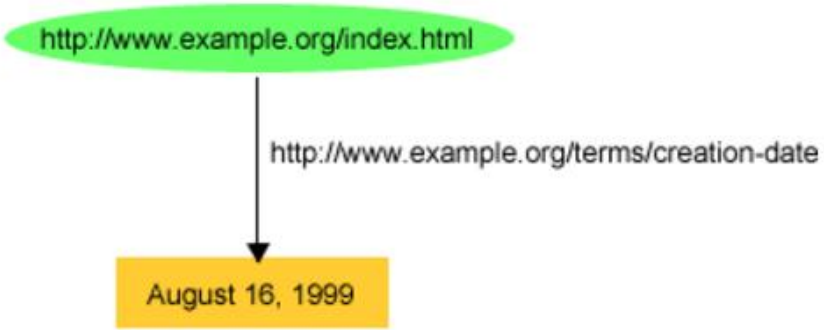
- Maintains semantics and is unambiguous
- Human and Machine processable vocabulary
- Triples!

Triples

- Terminology – Subject (Resource), Predicate (Property Type), Object (Value)

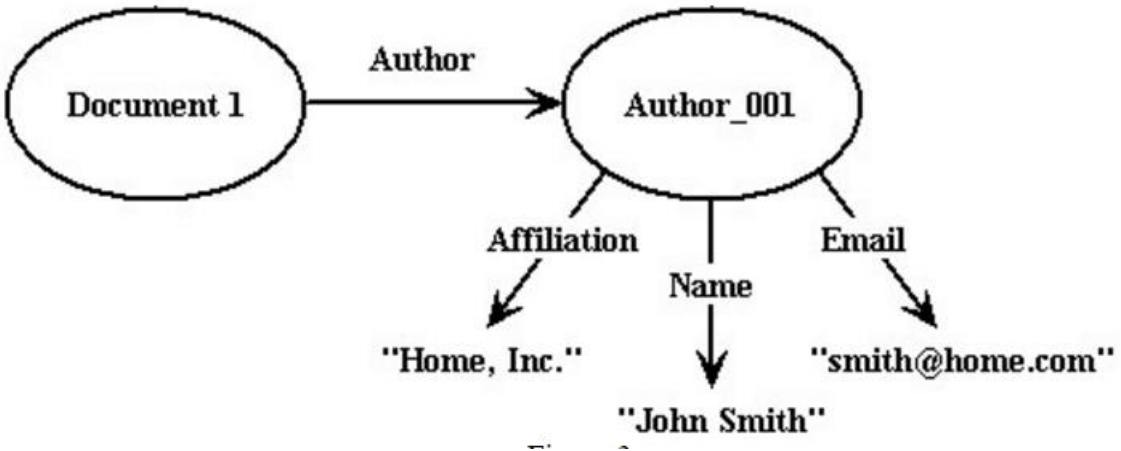
Simple RDF Statement

http://www.example.org/index.html has a **creation-date** whose value is **August 16, 1999**



source: https://www.w3.org/TR/2004/REC-rdf-primer-20040210/

RDF Graph



- To identify resources, RDF uses Uniform Resource Identifiers (URIs)

source: http://www.dlib.org/dlib/may98/miller/05miller.html

RDF/XML Syntax

RDF/XML for the Web Page's Creation Date

```
1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.             xmlns:exterms="http://www.example.org/terms/">

4.   <rdf:Description rdf:about="http://www.example.org/index.html">
5.       <exterms:creation-date>August 16, 1999</exterms:creation-date>
6.   </rdf:Description>

7. </rdf:RDF>
```

source: https://www.w3.org/TR/2004/REC-rdf-primer-20040210/

RDF Datatypes

- Datatype consists of:

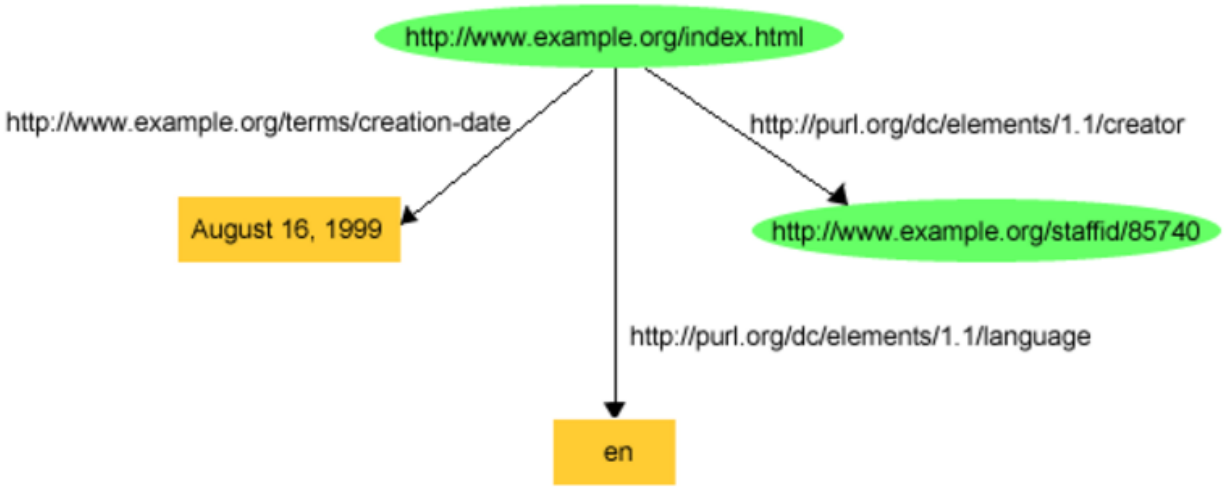
Value Space	{T, F}
Lexical Space	{"0", "1", "true", "false"}
Lexical-to-Value Mapping	{<"true", T>, <"1", T>, <"0", F>, <"false", F>}

- RDF predefines just one datatype – rdf:XMLLiteral

https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#section-Datatypes

RDF Literals

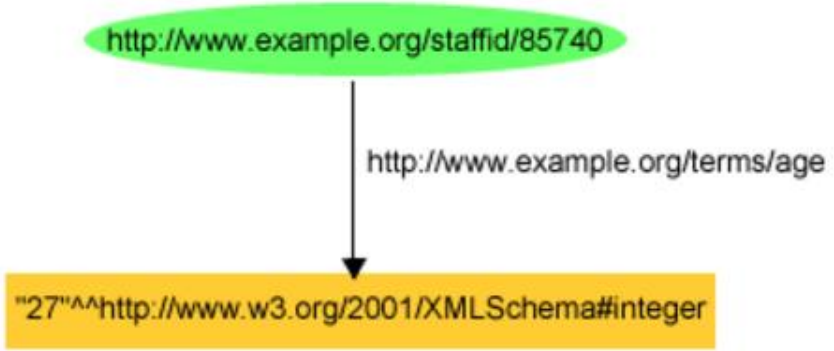
- Plain Literals - string combined with an optional language tag



https://www.w3.org/TR/2004/REC-rdf-primer-20040210/#rdfschema

RDF Literals

- Typed Literals - string combined with a datatype URI




```
<http://www.example.org/staffid/85740>  <http://www.example.org/terms/age> "27"^^<http://www.w3.org/2001/XMLSchema#integer> .
```

https://www.w3.org/TR/2004/REC-rdf-primer-20040210/#rdfschema

Operations

- Merge
- Flexible Schema
- No Key Constraints
- Cardinality

Triple Stores

- A database for storage and retrieval of RDF data
- Supports various query languages
  - SPARQL, RDQL, Versa
- Jena, Sesame, AllegroGraph, BigData

Why Triple Stores?

- Schema Flexibility
- Easy to Query
- Standards and Easy Data Migration
- Data Provenance

Why Triple Stores?

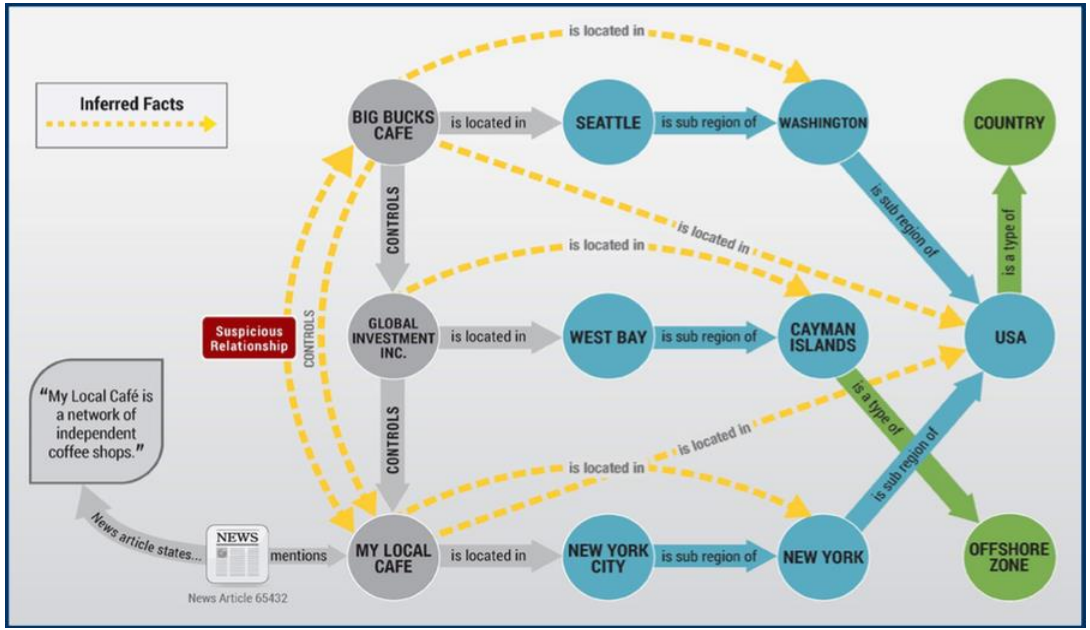
- Query Expressivity
  - No Joins, one table
- Example: Maria is a parent of Ivan
  Find tuples that are persons

Statement		
Subject	Predicate	Object
myo:Person	rdf:type	rdfs:Class
myo:gender	rdfs:type	rdfs:Property
myo:parent	rdfs:range	myo:Person
myo:spouse	rdfs:range	myo:Person
myd:Maria	rdf:type	myo:Person
myd:Maria	rdf:label	"Maria P."
myd:Maria	myo:gender	"F"
myd:Maria	rdf:label	"Ivan Jr."
myd:Ivan	myo:gender	"M"
myd:Maria	myo:parent	Myd:Ivan
myd:Maria	myo:spouse	myd:John
...		

source: https://ontotext.com/documents/white_papers/The-Truth-About-Triplestores.pdf

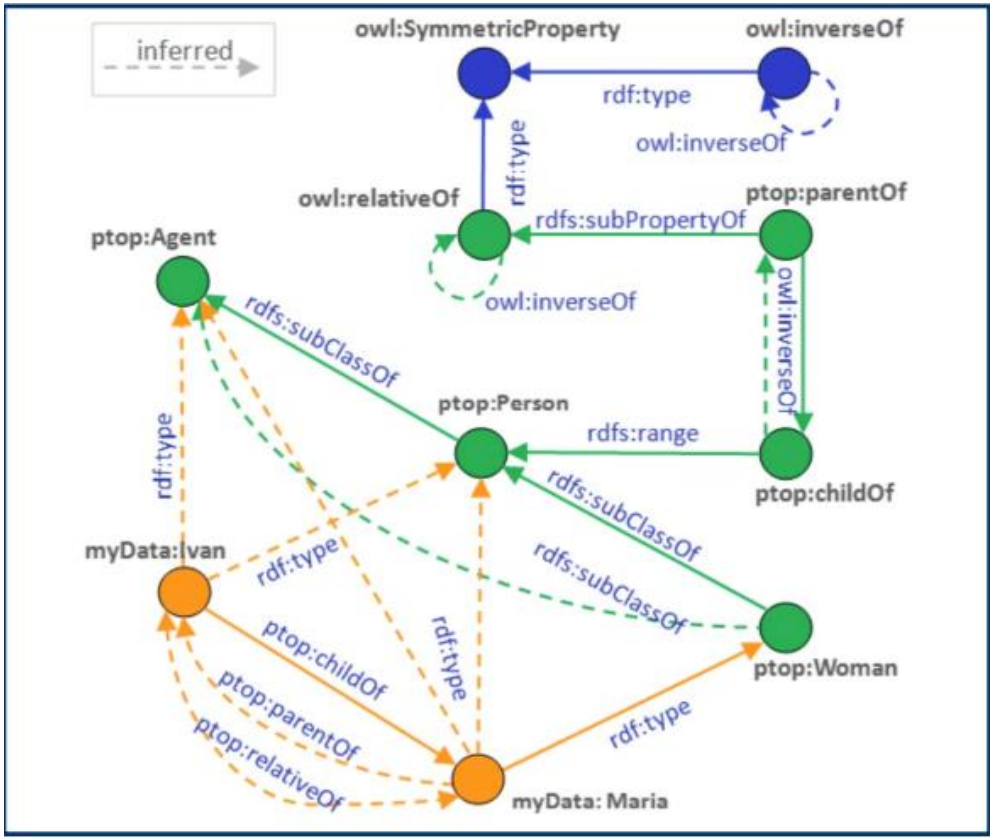
Why Triple Stores?

- Inferencing and Reasoning
  - Generate and infer new relationships among existing data.



source: https://ontotext.com/documents/white_papers/The-Truth-About-Triplestores.pdf

Inferencing and Reasoning



source: https://ontotext.com/documents/white_papers/The-Truth-About-Triplestores.pdf

Triple Store - Architecture

- Can be divided into three broad categories based on architecture of implementation:
  - In Memory
    - RDF graphs stored as triple in Main Memory
    - Fast, efficient but expensive

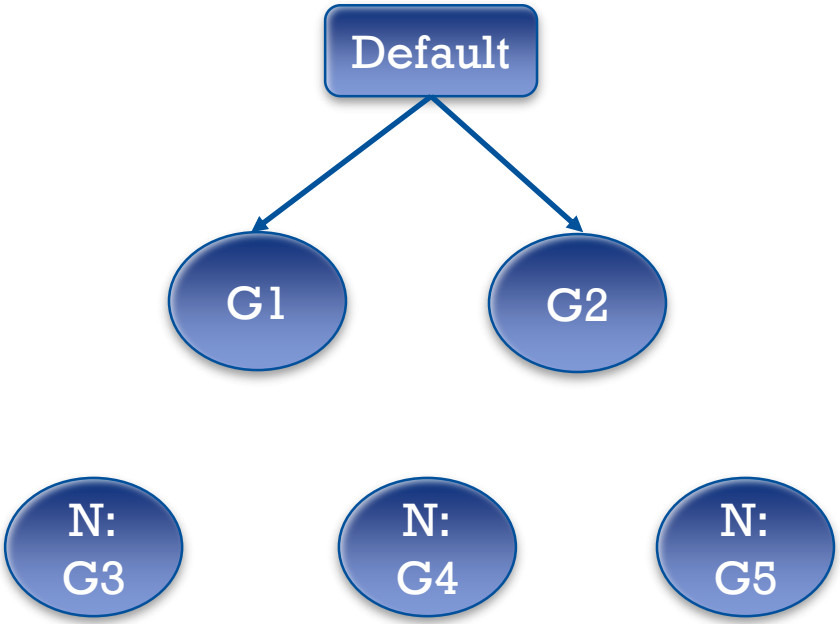
Triple Store - Architecture

- Native
  - Persistent storage with own implementation of the databases
  - Jena TDB, AllegroGraph etc.
- Non-Native
  - Run on third party databases like MySQL, Oracle etc.
  - Jena SDB

RDF Datasets

- Triple -> Graph -> Dataset
- Types
  - Default Graphs
  - Named Graphs
- Specified in SPARQL Queries

Default and Named Graphs



RDF Datasets – Another Example

exstaff:Sue	exterms:publication	ex:AnthologyOfTime
exstaff:Sue	exterms:publication	ex:ZoologicalReasoning
exstaff:Sue	exterms:publication	ex:GravitationalReflections
exstaff:Jack	exterms:publication	ex:Reasoning
exstaff:Jack	exterms:publication	ex:ThermalReflections
exstaff:Amy	exterms:publication	ex:Dimensionality
exstaff:Dan	exterms:publication	ex:SpaceandTime
exstaff:Bill	exterms:publication	ex:TheFourthDimension
exstaff:Kate	exterms:publication	ex:AtmosphericSignalInterference

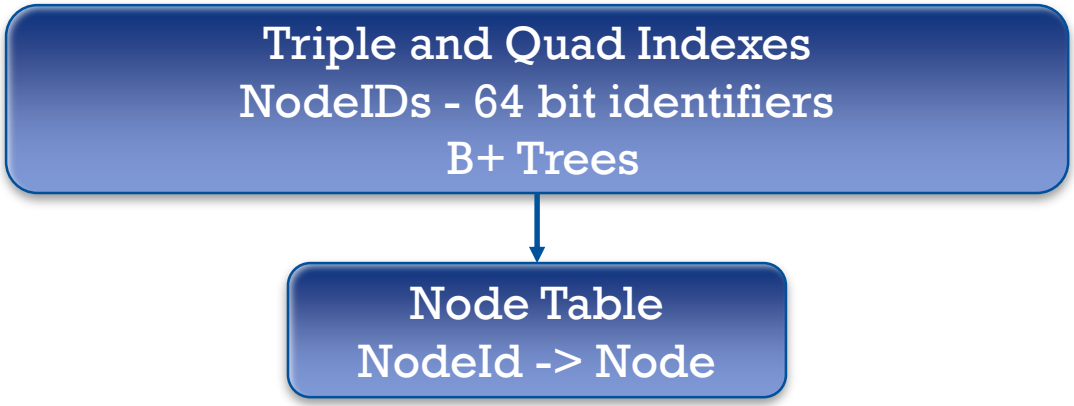
```
SELECT ?a
FROM <ex1.ttl>
FROM <ex2.ttl>
FROM NAMED <ex3.ttl>
WHERE{
        { ?b publication ?a }
UNION { GRAPH <ex3.ttl> { ?b publication ?a }
}
}
```

Jena SDB

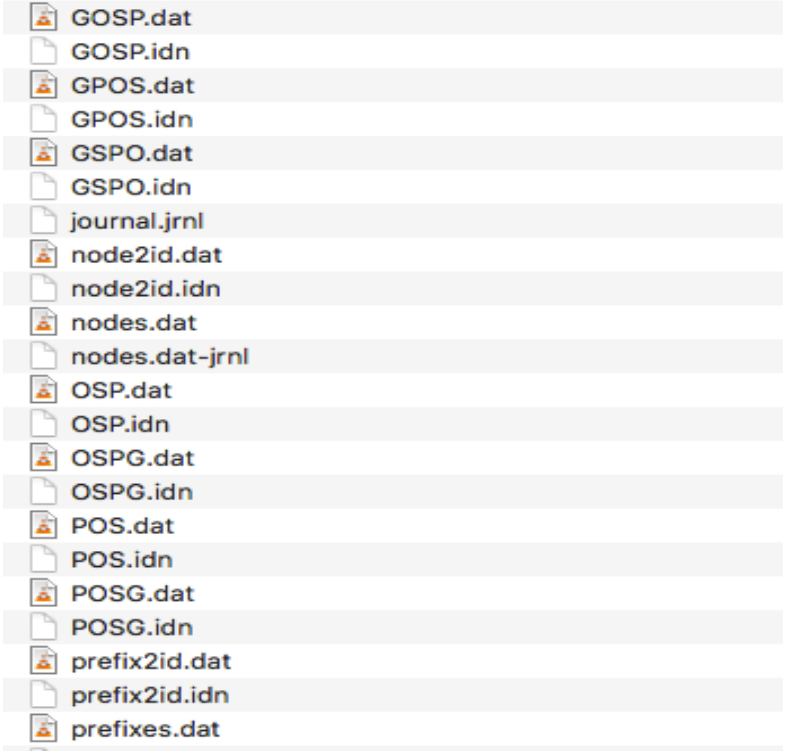
- Old implementation of Triple Stores.
- Relational database for storage and querying of RDF data
- Multiple stores supported: MySQL, PostgreSQL, Oracle, DB2

Jena TDB

- Component of Jena for RDF storage and query
- Stored in a single directory



Jena TDB


GOSP.dat
GOSP.idn
GPOS.dat
GPOS.idn
GSPO.dat
GSPO.idn
journal.jrnl
node2id.dat
node2id.idn
nodes.dat
nodes.dat-jrnl
OSP.dat
OSP.idn
OSPG.dat
OSPG.idn
POS.dat
POS.idn
POSG.dat
POSG.idn
prefix2id.dat
prefix2id.idn
prefixes.dat

Jena TDB

- Managed through Command Line and Jena API
- ACID properties through WRITE-AHEAD-LOGGING
- Provides Serializable transactions
- Queried using SPARQL

RDF Data as Relational Database

	Subject	Predicate	Object
triple →	Primary Key		(heterogeneous)
→			
→			
→			

QUERY ENGINE - ARQ

QUERY LANGUAGE - SPARQL

Query Engine - ARQ

- Supports SPARQL RDF Query language
- Offers free text search via Lucene
- Access and extension of SPARQL algebra
- Support for Remote Federated Queries
- Extension to other storage systems

SPARQL

- SPARQL Protocol and RDF query language
- Queries consist of triple patterns (triple store)
- Allow analytic operations like JOIN, SORT, AGGREGATE
- Query unknown relationships
- Pull values from structured and semi-structured data

SPARQL as SQL

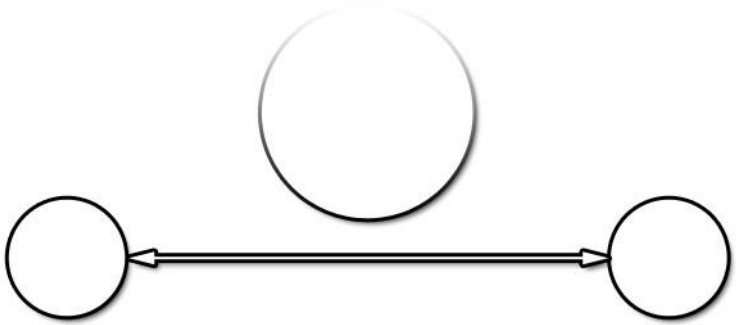
- SPARQL : Semantic Web : : SQL : Relational Databases
  - 3 columns – subject, predicate, object
  - Add new predicates without changing schema

- SPARQL : Resources : : SQL : Tables and Databases
  - Derive any information!
  - Draw conclusions and make predictions!

SPARQL Query comprises

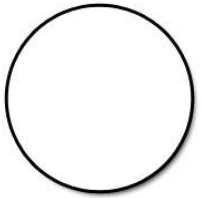
- Prefix declarations – BASE, PREFIX
- Dataset definition – FROM, FROM NAMED
- Result Clause – SELECT, CONSTRUCT, DESCRIBE, ASK
- Query pattern – WHERE
- Query modifiers – ORDER BY, LIMIT, DISTINCT, REDUCED

Dataset Specification Scheme

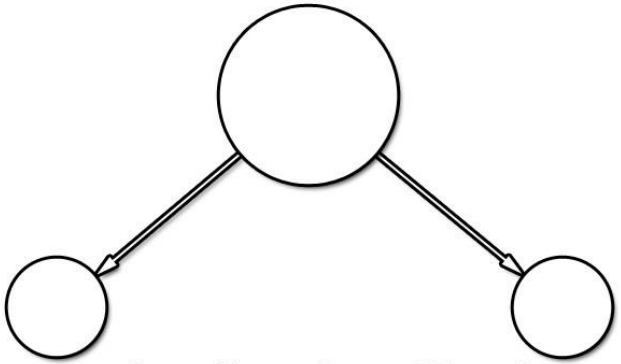


FROM NAMED <http://example.org/alice>

FROM NAMED <http://example.org/bob>



FROM <http://example.org/dft.ttl>



FROM <http://example.org/dft.ttl>

FROM NAMED <http://example.org/alice>

FROM NAMED <http://example.org/bob>

Source: https://data-gov.tw.rpi.edu/wiki/How_to_specify_RDF_Dataset_in_SPARQL_queries

QUERY FORMS

- SELECT query
- CONSTRUCT query
- ASK query
- DESCRIBE query

SPARQL query processing stages

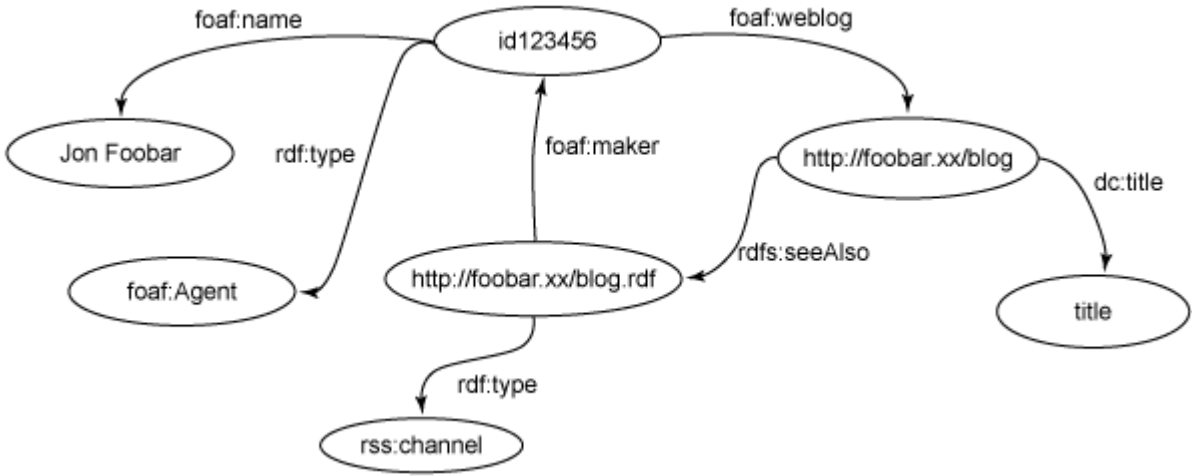
- Parsing - String to Query
- Translation - Query to SPARQL algebra expression
- Optimization of algebra expression
- Determination of query plan
- Evaluation of query plan

Basic SPARQL query

"Find the URL of the blog by the person named Jon Foobar"

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?url
FROM <bloggers.rdf>
WHERE {
        ?contributor foaf:name “Jon Foobar” .
        ?contributor foaf:weblog ?url .
}
```

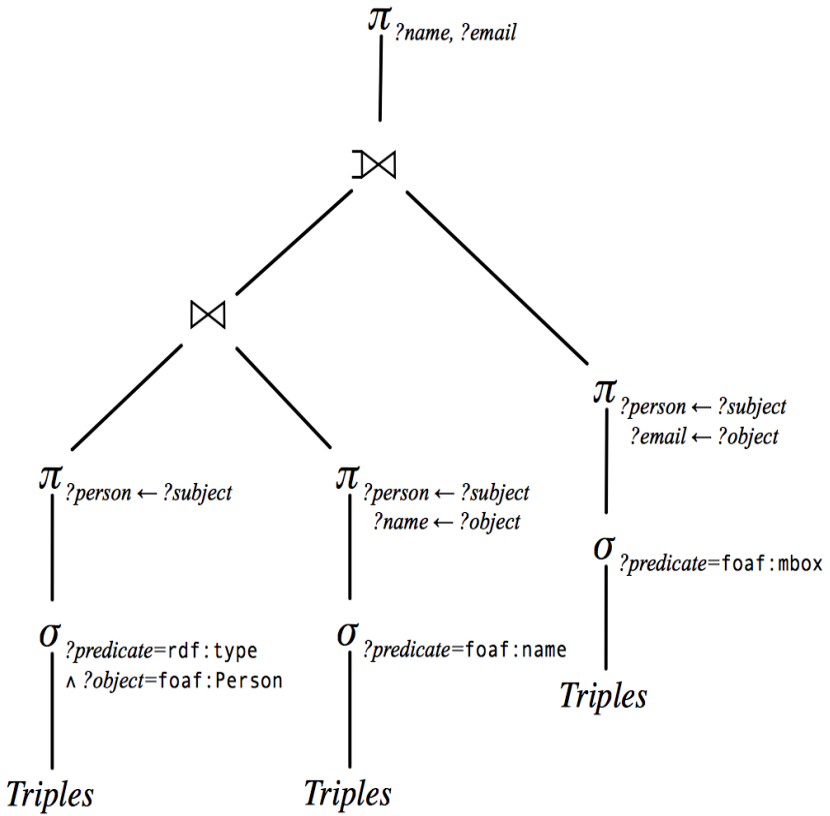
Anatomy of a SPARQL query



Source: http://www.ibm.com/developerworks/xml/library/j-sparql/

Relational Algebra for SPARQL

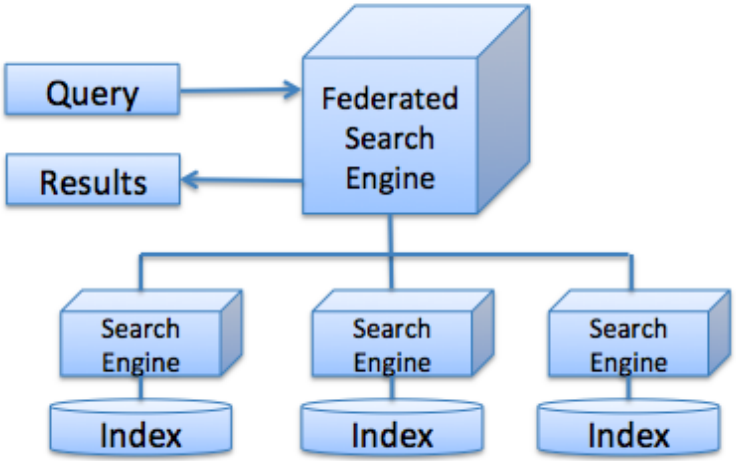
```
SELECT ?name ?email
WHERE {
?person rdf:type foaf:Person .
?person foaf:name ?name .
OPTIONAL { ?person foaf:mbox ?email
} }
```



Source: http://www.hpl.hp.com/techreports/2005/HPL-2005-170.pdf

Federated Search

- Search multiple resources simultaneously
- Distribute query, aggregate results
- Disparate databases
- Wrapper function for translation



Source: https://en.wikipedia.org/wiki/Federated_search

Advantages of SPARQL

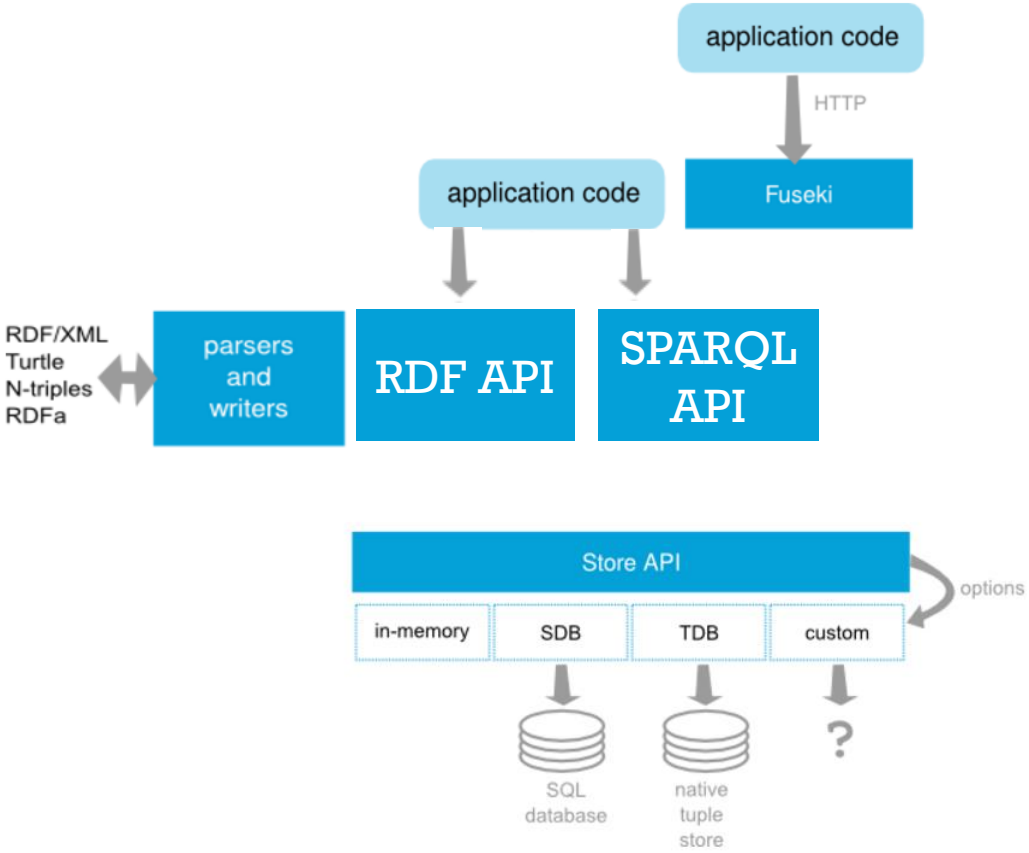
- Schema Flexibility, no downtime or redesign
- Less Development Time, low cost
- Federating information from websites and databases
- Complex joins of structured and semi-structured data
- Good interoperability with other software systems

Limitations of SPARQL

- Immaturity – lack of wide deployment
- Predicates cannot have properties
- Negation is complicated
- Lacks support for transitive/hierarchical queries

Implementation

- JENA Core API
- SPARQL Query on Database
- SPARQL Via API



Source: https://jena.apache.org/about_jena/architecture.html

JENA Core API

- RDF Graph = Model
- Triples = Statement
- Namespaces and curie

```
// Create an empty Model
Model model = ModelFactory.createDefaultModel();
```

```
Statement statement = model.createStatement(adam,parentOf,fran);
```

SPARQL Command Line

- Using ARQ

```
$ sparql --query jon-url.rq

---------------------------
| url                     |
===========================
| <http://foobar.xx/blog>  |
---------------------------
```

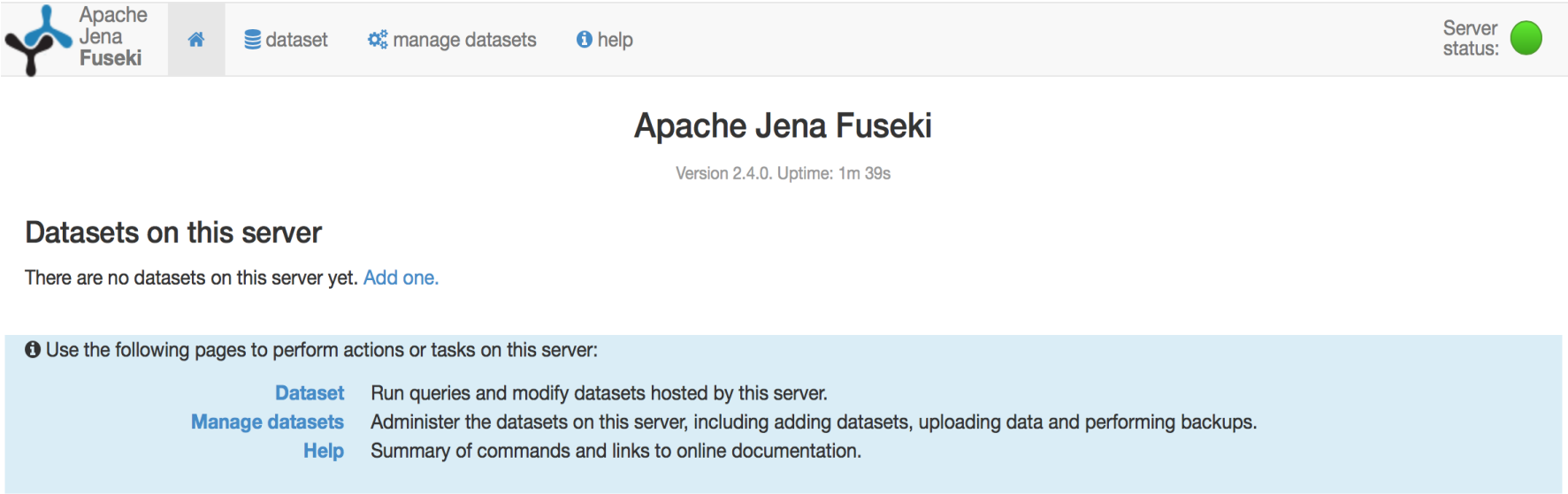
- sparql --data URL

SPARQL With JENA API

- Similar to JDBC
- QueryFactory create()
- QueryExecution
- execSelect()

FUSEKI

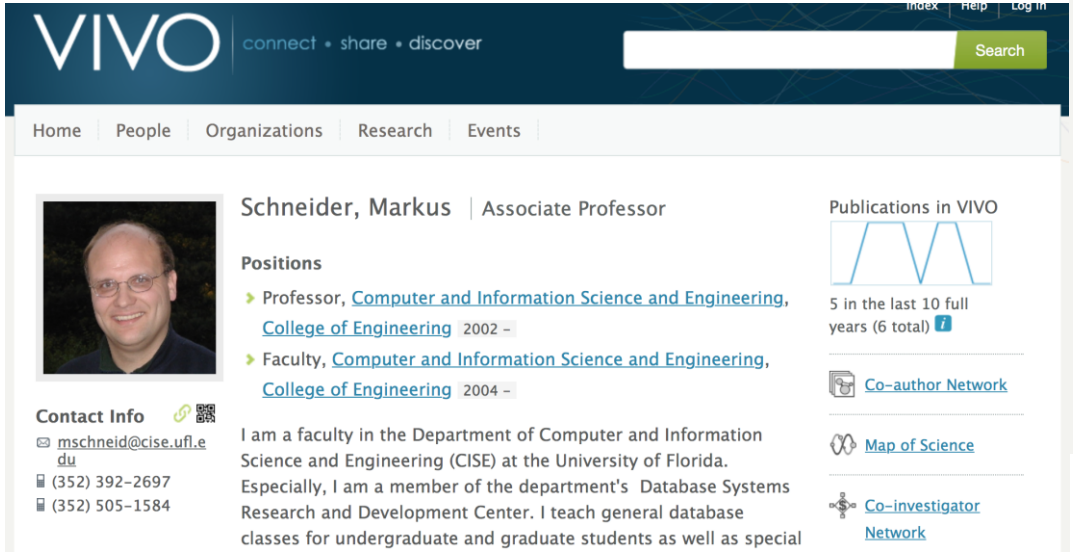
- SPARQL Server



Features

- Scalability
- Serialization
- Inference

Applications



has course role

CIS4301 Info & Database Sys 1

CIS6905 Individual Study

CIS6910 Supervised Research

CIS6930 Special Topics

CIS6930 Special Topics

... more

Affiliation

home department

Computer and Information Science and Engineering

Source: http://vivo.ufl.edu

Applications

- Fedora - Flexible Extensible Digital Object Repository Architecture
- GRANATUM – Web Portal for Drug Discovery
- UK Government – Land Registry
- Environment Data

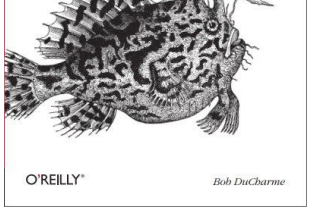
Source: http://www.topquadrant.com/company/customers/

Limitations

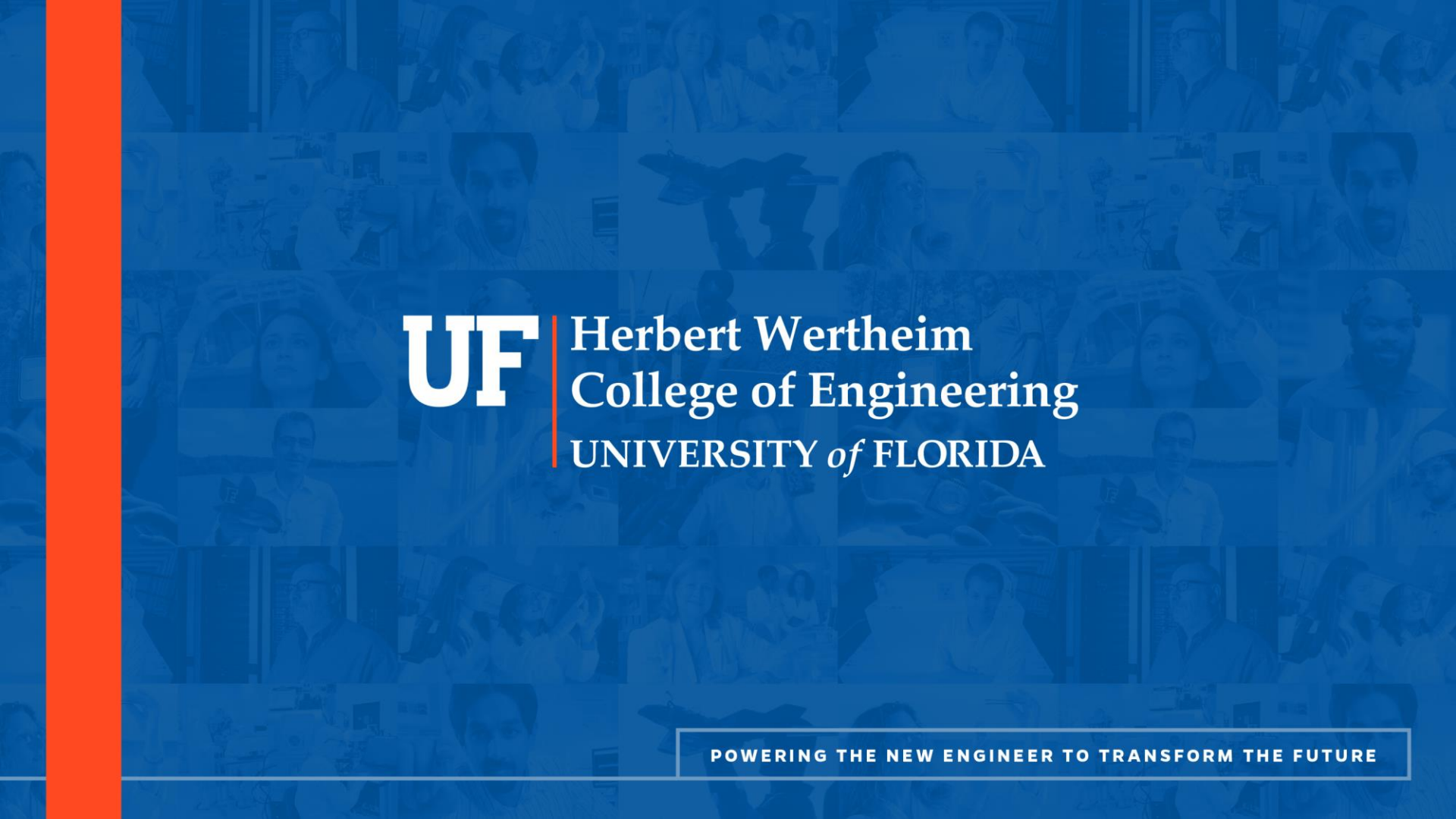
- Limited API
- Centralized

Further Reading

- RDF Primer, W3C
- Apache Jena
- Learning SPARQL, Bob DuCharme



Questions


UF
Herbert Wertheim
College of Engineering
UNIVERSITY of FLORIDA
POWERING THE NEW ENGINEER TO TRANSFORM THE FUTURE