Graph Database Model supporting RDF

Renzo Angles

Department of Computer Science
Universidad de Chile

AMW'07

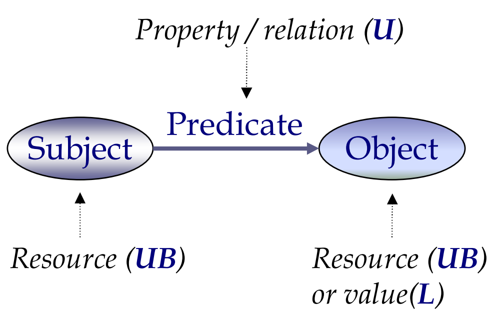
General objective

- Call the attention of the RDF community about related work in (graph) databases
- Use the experience of the database community to enhance research on RDF

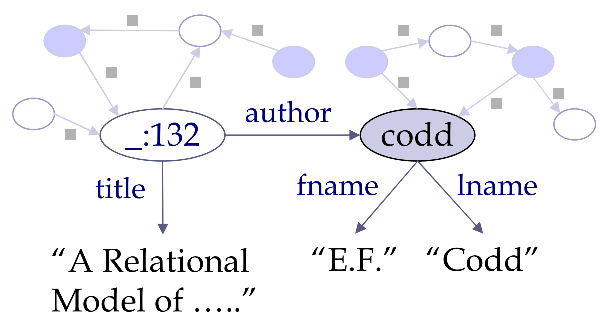
RDF Data Model (1999)

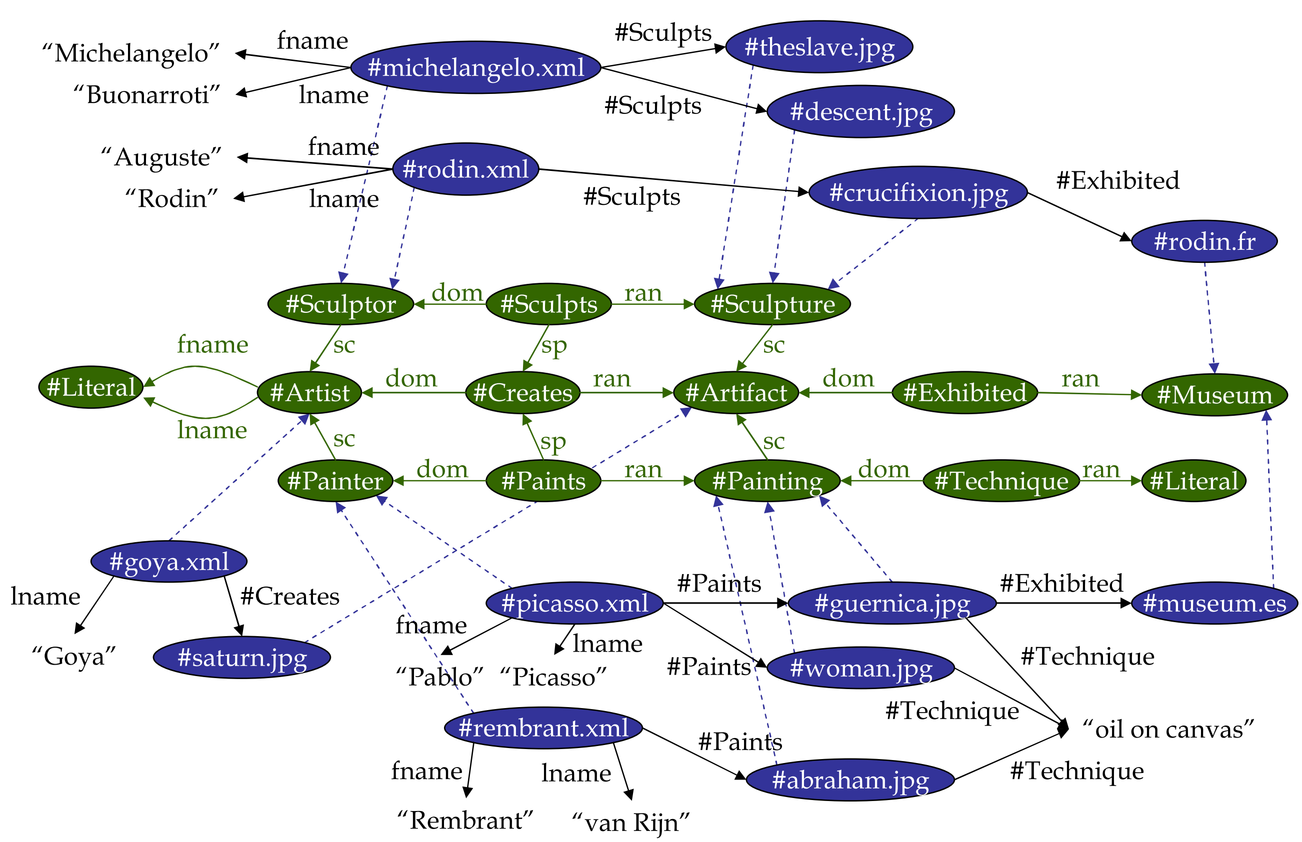
- Domains (**U**B**L**): Resources (**U**RIs), **B**lank nodes (existencial variables), **L**iterals
- Data structures:

RDF Triple



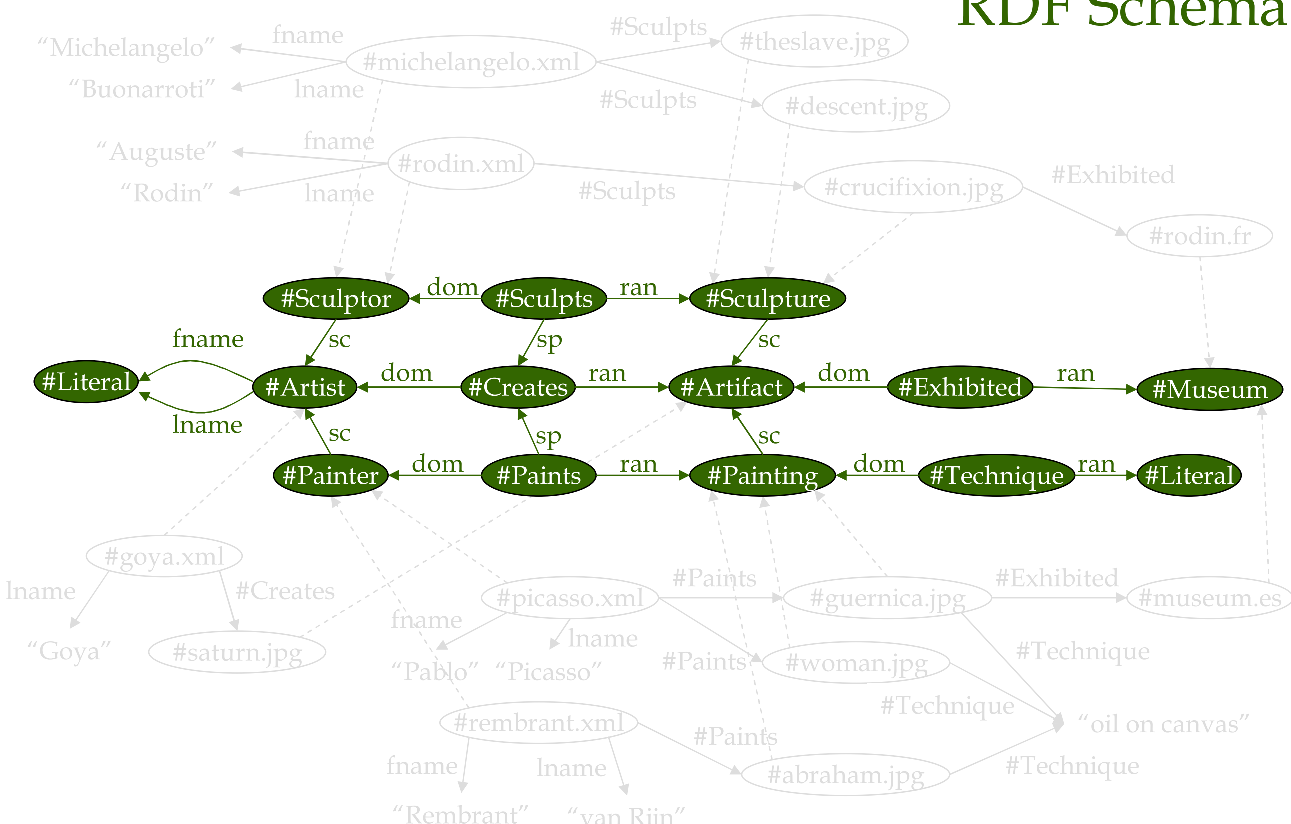
RDF Graph





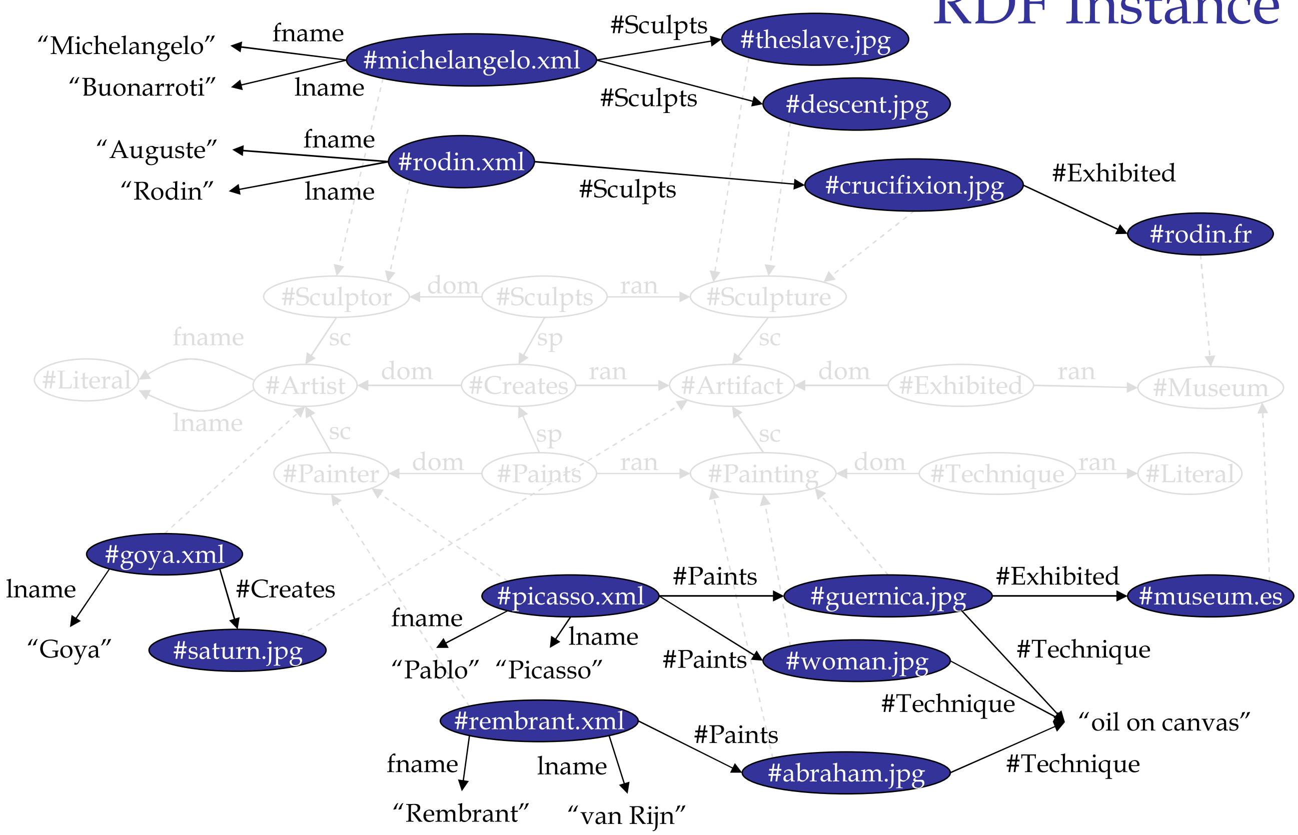
"Michelangelo"
fname
#michelangelo.xml
#Sculpts
#theslave.jpg
"Buonarroti"
lname
#Sculpts
#descent.jpg
"Auguste"
fname
#rodin.xml
"Rodin"
lname
#Sculpts
#crucifixion.jpg
#Exhibited
#rodin.fr
#Sculptor
dom
#Sculpts
ran
#Sculpture
fname
sc
sp
sc
#Literal
#Artist
dom
#Creates
ran
#Artifact
dom
#Exhibited
ran
#Museum
lname
sc
sp
sc
#Painter
dom
#Paints
ran
#Painting
dom
#Technique
ran
#Literal
#goya.xml
lname
#Creates
"Goya"
#saturn.jpg
#picasso.xml
#Paints
#guernica.jpg
#Exhibited
#museum.es
fname
lname
"Pablo"
"Picasso"
#Paints
#woman.jpg
#Technique
#Technique
#rembrant.xml
#Paints
"oil on canvas"
fname
lname
#abraham.jpg
#Technique
"Rembrant"
"van Rijn"

RDF Schema

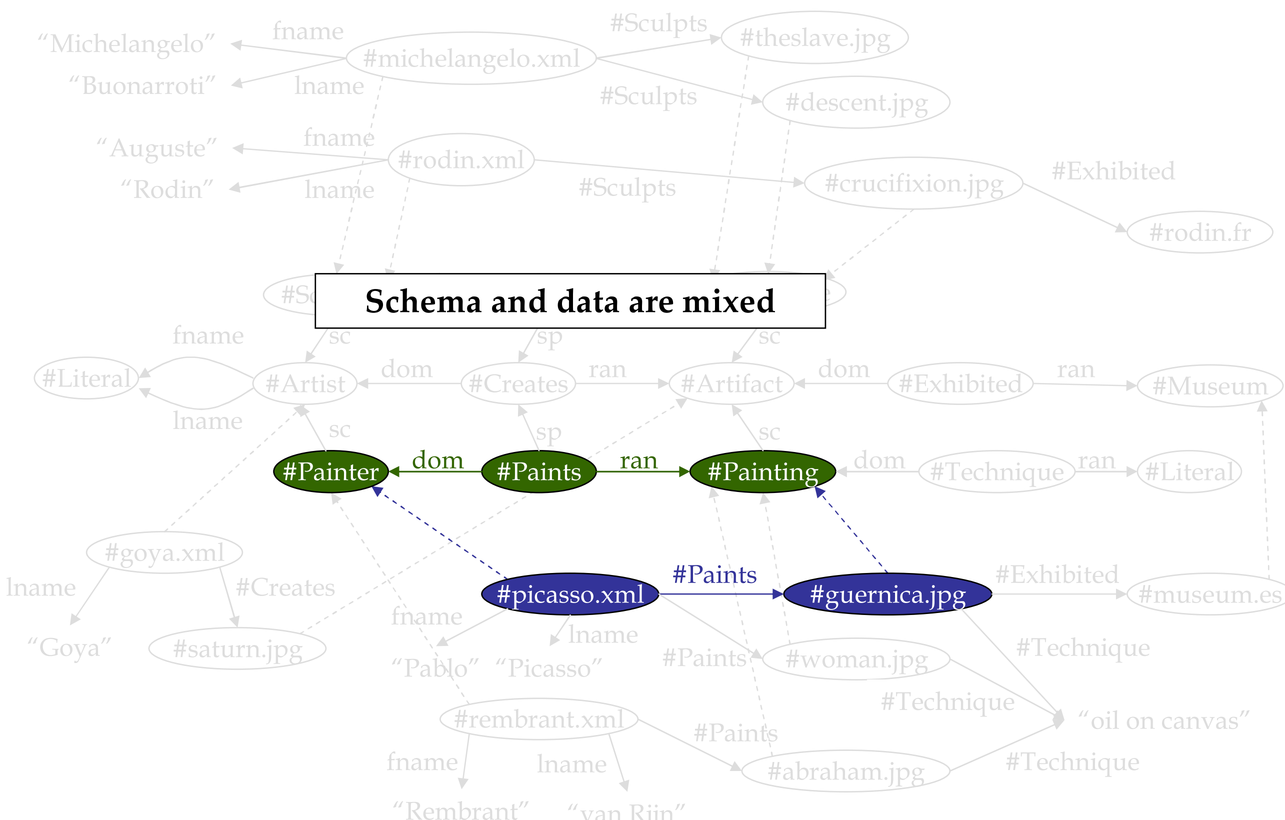


"Michelangelo"
fname
#michelangelo.xml
#Sculpts
#theslave.jpg
"Buonarroti"
lname
#Sculpts
#descent.jpg
"Auguste"
fname
#rodin.xml
"Rodin"
lname
#Sculpts
#crucifixion.jpg
#Exhibited
#rodin.fr
#Sculptor
dom
#Sculpts
ran
#Sculpture
fname
sc
sp
sc
#Literal
#Artist
dom
#Creates
ran
#Artifact
dom
#Exhibited
ran
#Museum
lname
sc
sp
sc
#Painter
dom
#Paints
ran
#Painting
dom
#Technique
ran
#Literal
#goya.xml
lname
#Creates
#picasso.xml
#Paints
#guernica.jpg
#Exhibited
#museum.es
"Goya"
#saturn.jpg
fname
lname
"Pablo"
"Picasso"
#Paints
#woman.jpg
#Technique
#Technique
#rembrant.xml
#Paints
"oil on canvas"
fname
lname
#abraham.jpg
#Technique
"Rembrant"
"van Rijn"

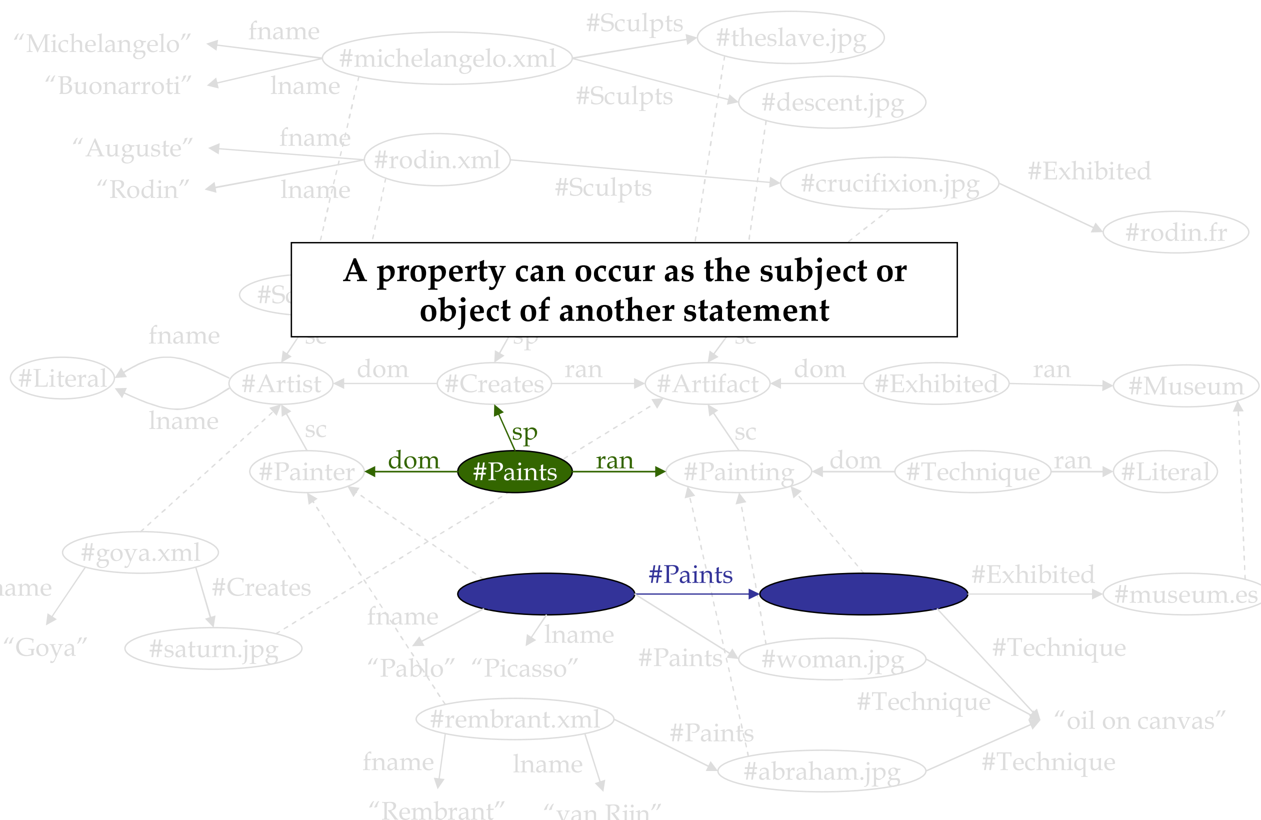
RDF Instance



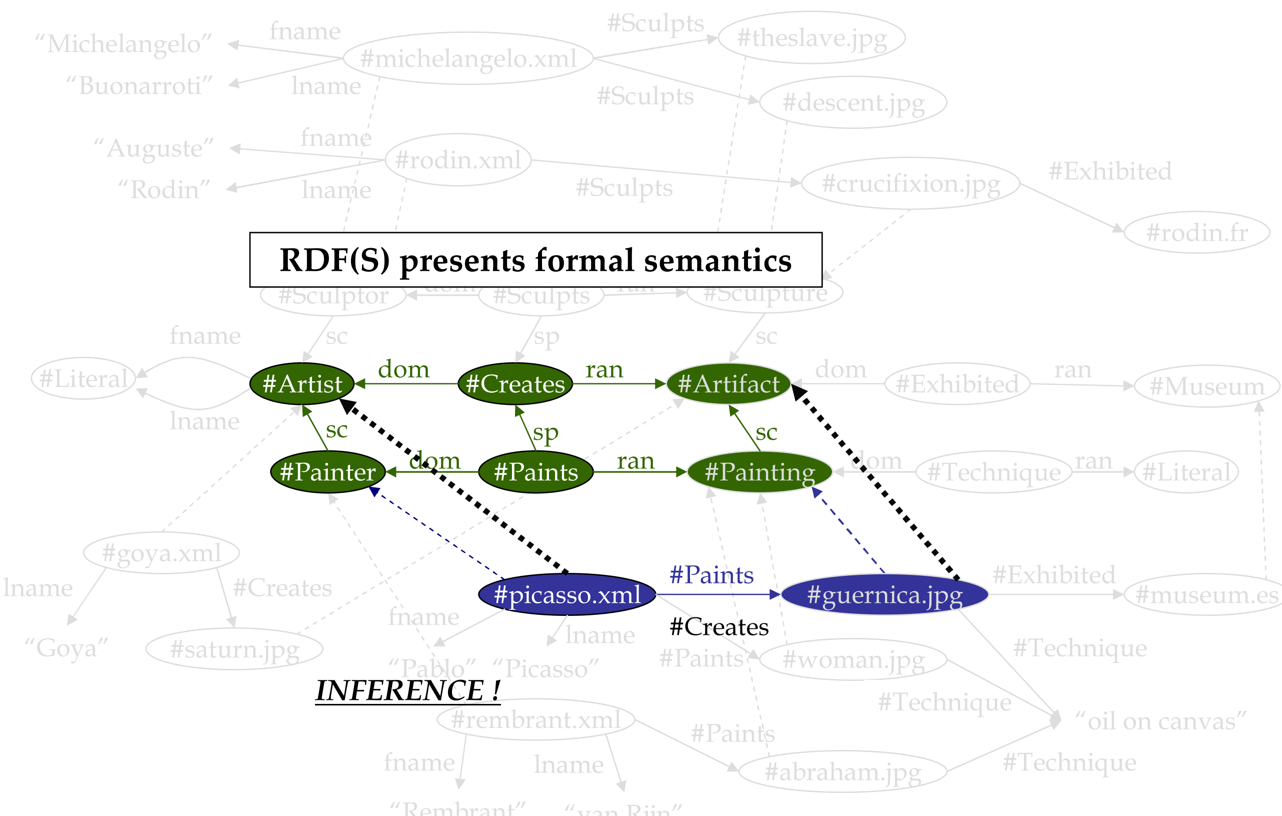
RDF(S) model features



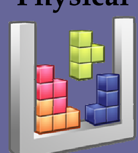
RDF(S) model features



RDF(S) model features



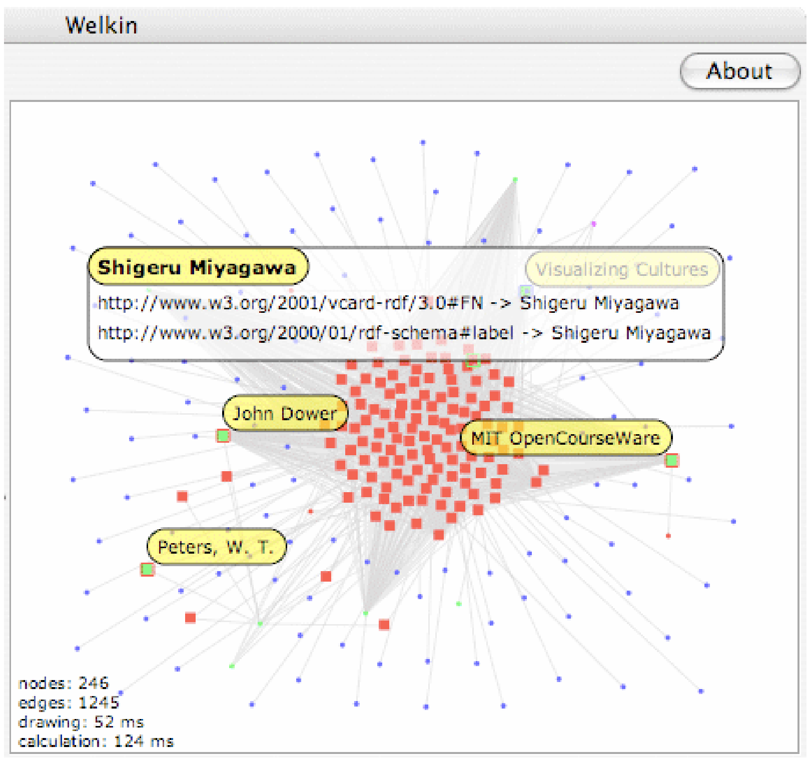
RDF from a database perspective

	RDF API	RDF REPOSITORY	RDF DATABASE
View 	Functions	RDF QL (SPARQL) Rule-based Inference engine	?
Logical 	Triples	Triples RDF Graphs Datasets (SPARQL)	?
Physical 	File	Native data store Files RDBMS	?
Abstraction Level			

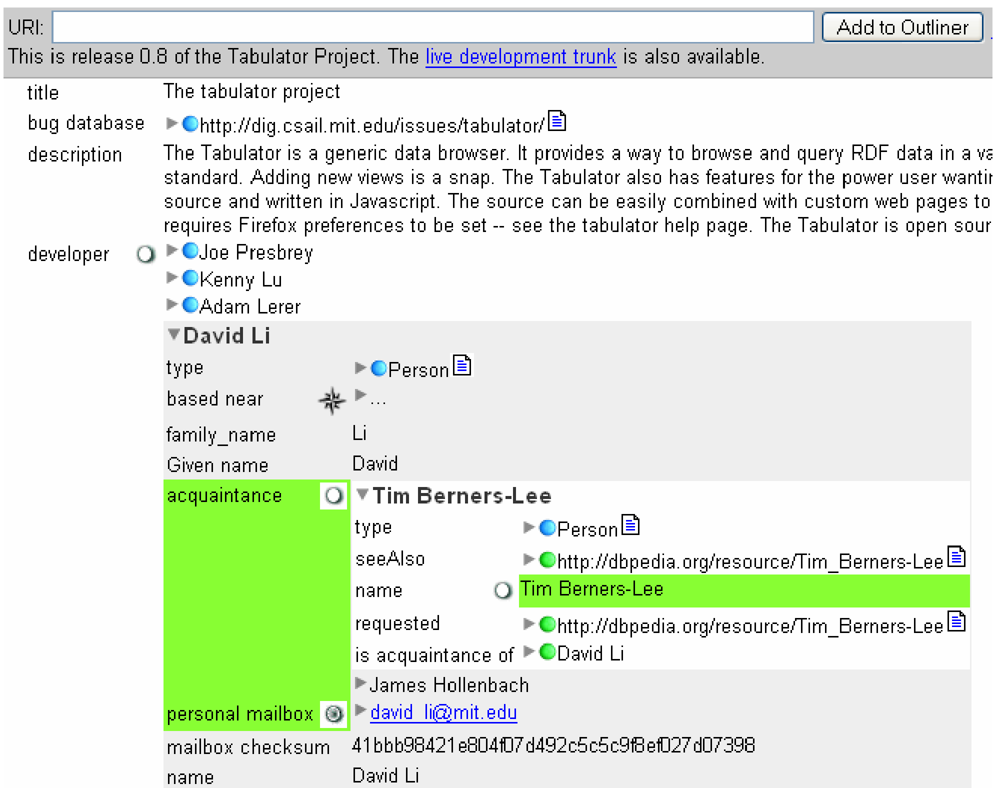
Issues from a database perspective

- The RDF model has a very low level of abstraction
- RDF query languages need support for graph properties
- RDF does not define (formally) notions of integrity constraints
- Improve RDF visualization
- Use of graph data structures and algorithms for secondary memory

Welkin: graph-based RDF visualizer



Tabulator: semantic data browser



Tabulator: visualizing, browsing and querying of RDF data

Tabulator: querying and browsing

dar⊠ Timeline⊠ SPARQL⊠ New...

```
SELECT ?v0 ?v1 ?v2 ?v3
WHERE
{
    <http://dig.csail.mit.edu/2005/ajar/ajaw/data#Tabulator>
<http://usefulinc.com/ns/doap#developer> ?v0 .
    ?v0 <http://xmlns.com/foaf/0.1/knows> ?v1 .
    ?v1 <http://xmlns.com/foaf/0.1/name> ?v2 .
    FILTER ( ?v2 = "Tim Berners-Lee" )
    OPTIONAL {
        ?v0 <http://xmlns.com/foaf/0.1/mbox> ?v3 .
    }
}
```

Table⊠ Map⊠ Calendar⊠ Timeline⊠ SPARQL⊠ New...

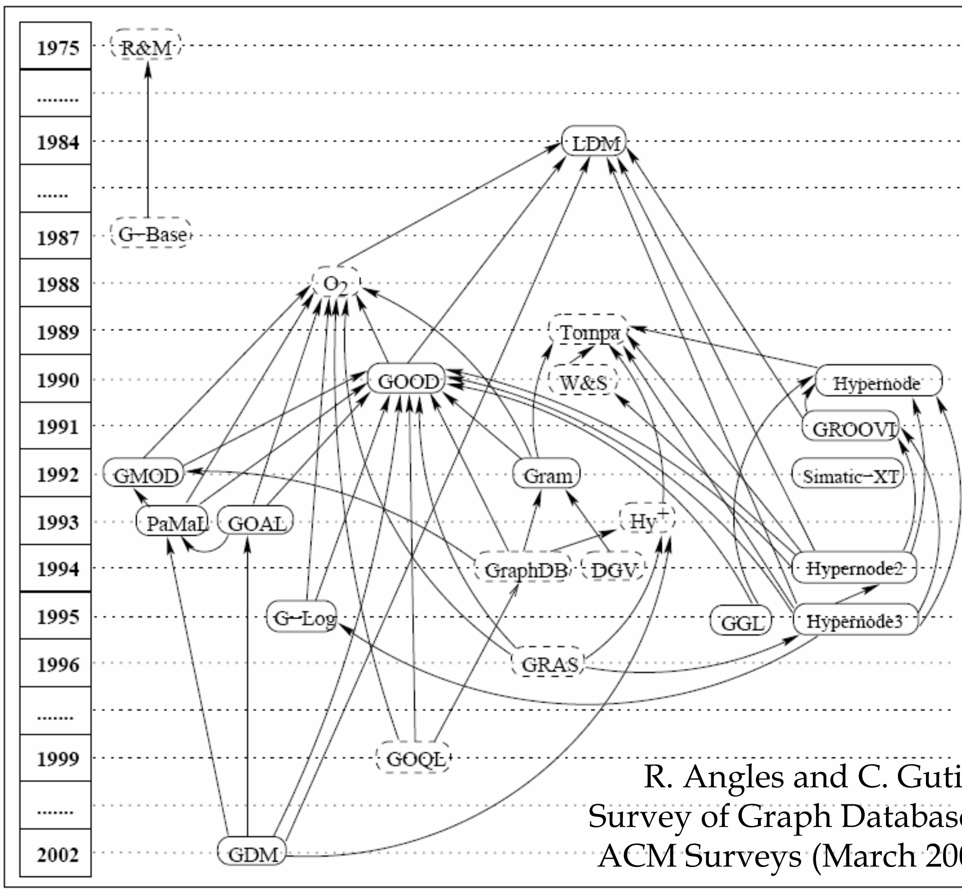
Select queries to display: ⊠

Query #1 ×

The tabulator project developer	The tabulator project developer acquaintance	The tabulator project developer acquaintance name	The tabulator project developer personal mailbox
David Li	Tim Berners-Lee	Tim Berners-Lee	mailto:david_li@mit.edu
Sonia Nijhawan	Timothy Berners-Lee	Tim Berners-Lee	.

RDF from a database perspective

	RDF API	RDF REPOSITORY	RDF DATABASE
View 	Functions	RDF QL (SPARQL) Rule-based Inference engine	(Graph) Query Language Rule-based inference engine RDF Integrity Constraints (Rule-definition language)
Logical 	Triples	Triples RDF Graphs Datasets (SPARQL)	(Graph) Database Model
Physical 	File	Native data store Files RDBMS	Native data store RDBMS?
Abstraction Level			



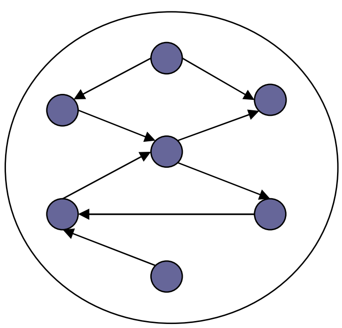
R. Angles and C. Gutierrez
Survey of Graph Database Models
ACM Surveys (March 2008 issue)

Graph, Hypernode, Hypergraph

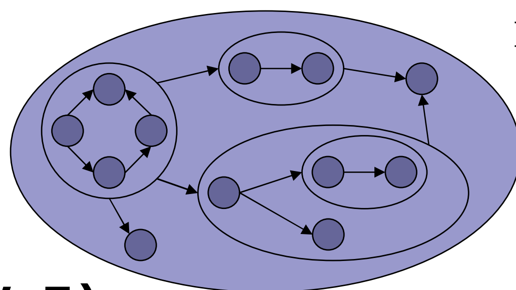
N = set of simple nodes   **H** = set of hypernodes

Graph

V⊆N   E⊆VxV

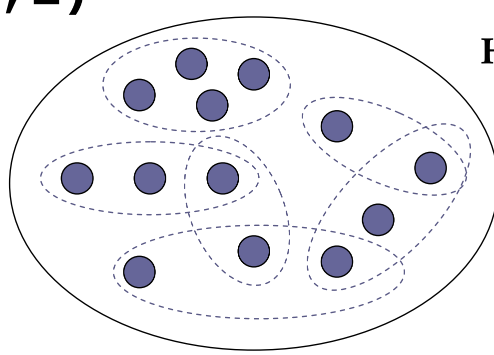


G =(V,E)



Hypernode

V = N∪H

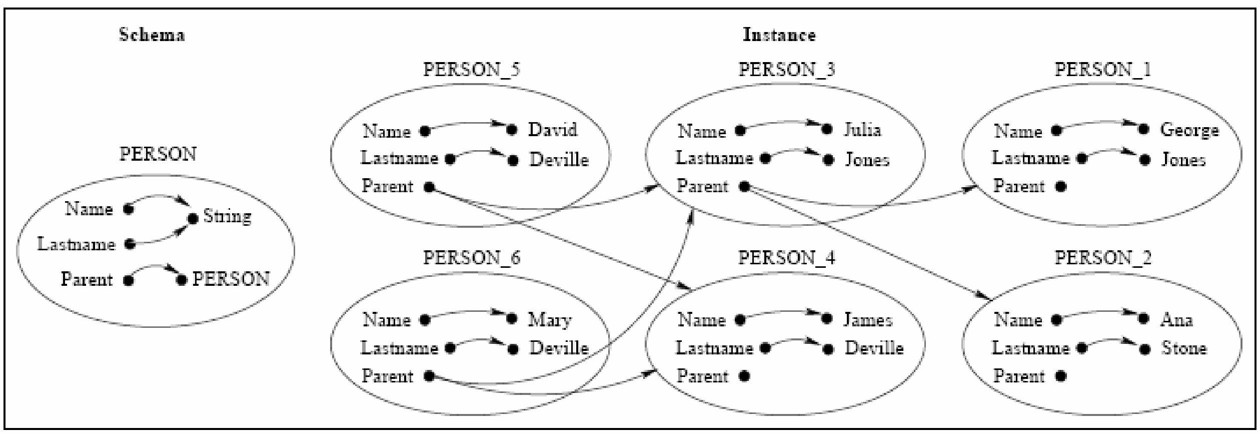


Hypergraph

E⊆P(V)

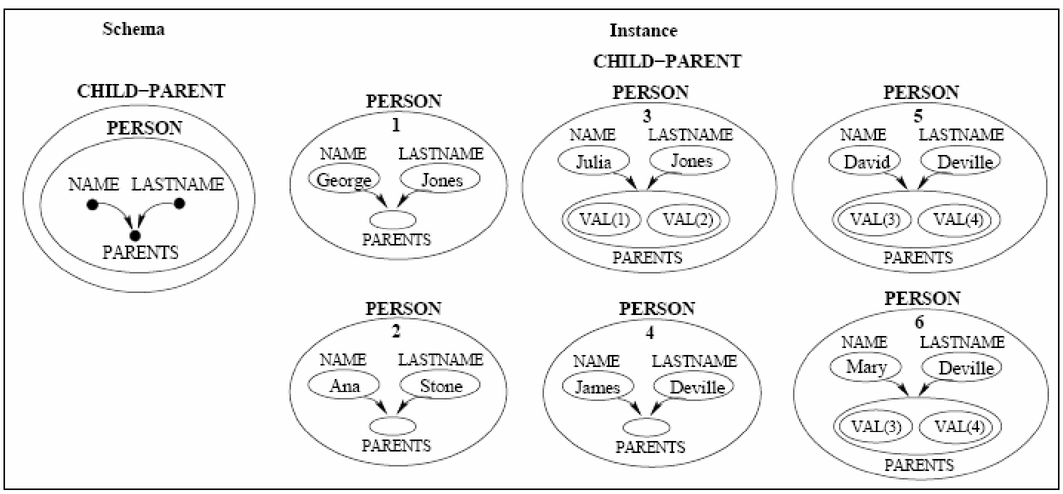
Hypernode Model (1990)

- **Motivation:** modeling of complex objects
- **Features:** simple and flexible data structure (***hypernode***) that supports complex objects and encapsulation of information

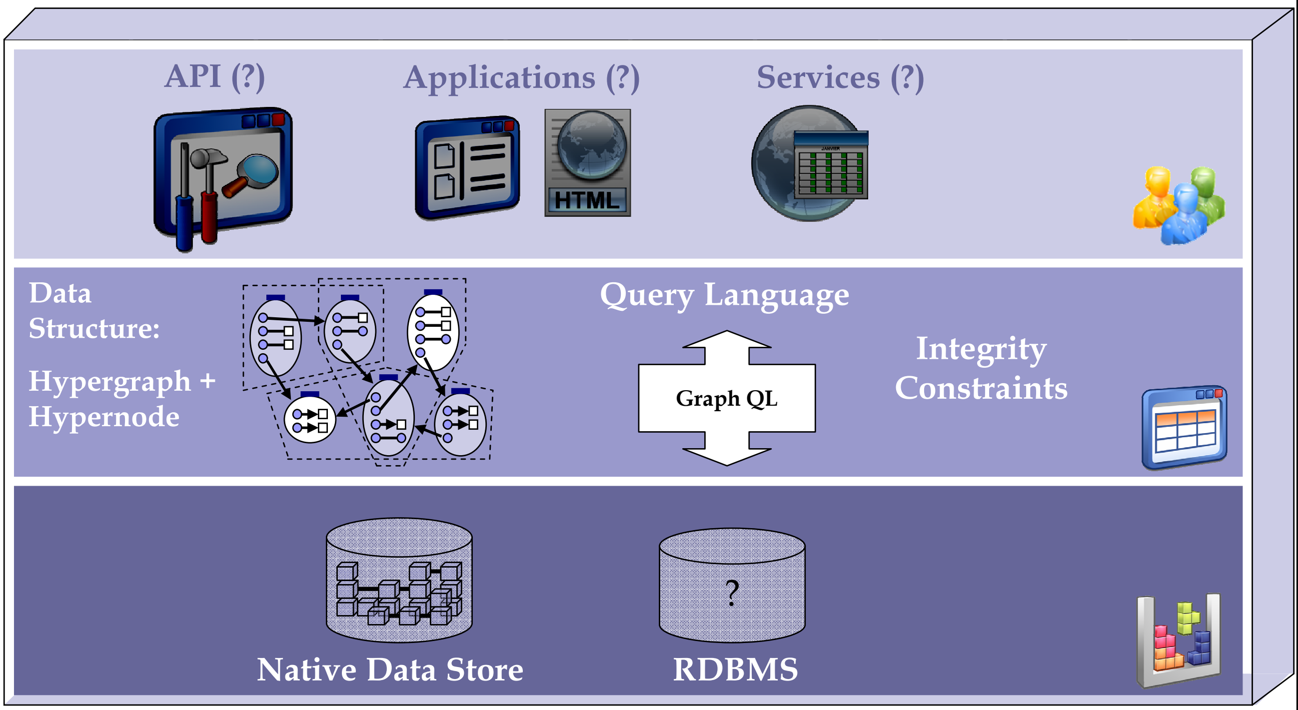


Hypergraph Model (1991)

- **Motivation:** modeling of object-oriented features, visualization, browsing
- **Features:** natural formalisation of the notions of sub-object sharing and structural inheritance



Proposal


API (?)
Applications (?)
HTML
Services (?)
Data
Structure:
Hypergraph +
Hypernode
Query Language
Graph QL
Integrity
Constraints
?
Native Data Store
RDBMS

Graph Schema

A *Graph Schema* is a hypergraph (V, E_d, E_u), where

- $V \subset U$ is a set of vertices
- $E_d \subseteq V \times V$ is a set of unlabeled directed edges (just edges)
- E_u is a set of labeled undirected hyperedges (just hyperedges) of the form (c, P) where:
  - $c \in U$ is the hyperedge label
  - $P \subseteq E_d$ is the set of edges grouped by the hyperedge

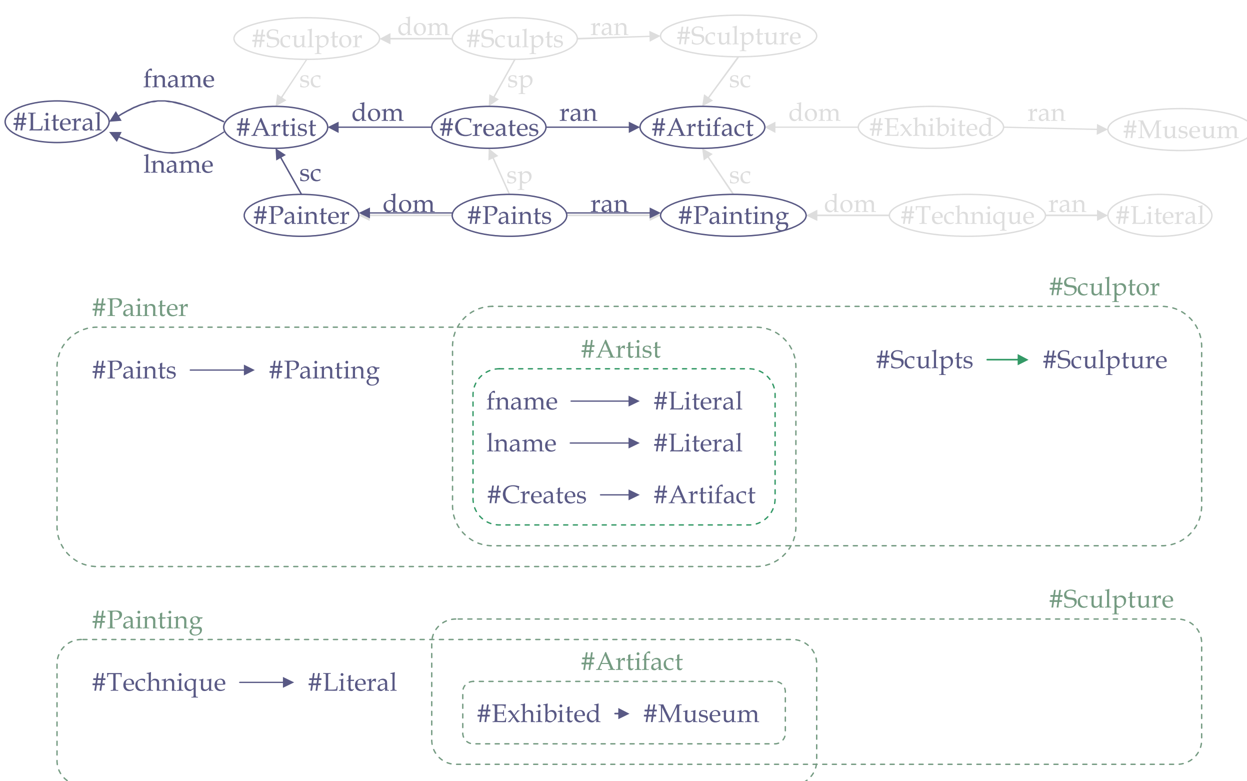
Schema for Resources (definition)

A *Resource Graph Schema* is a graph schema (V, E_d, E_u) with the following interpretation:

- A hyperedge $(c, P) \in E_u$ represents a *Resource Class* where:
  - c is the name of the class
  - Each edge $(u, v) \in P$ represents a *Property* as a pair *(propery-name,property-value)*.
- If a resource class c_1 contains a resource class c_2, then c_1 is a *subClassOf* c_2 (i.e. inheritance of properties)

Schema for Resources (example)

Hyperedge

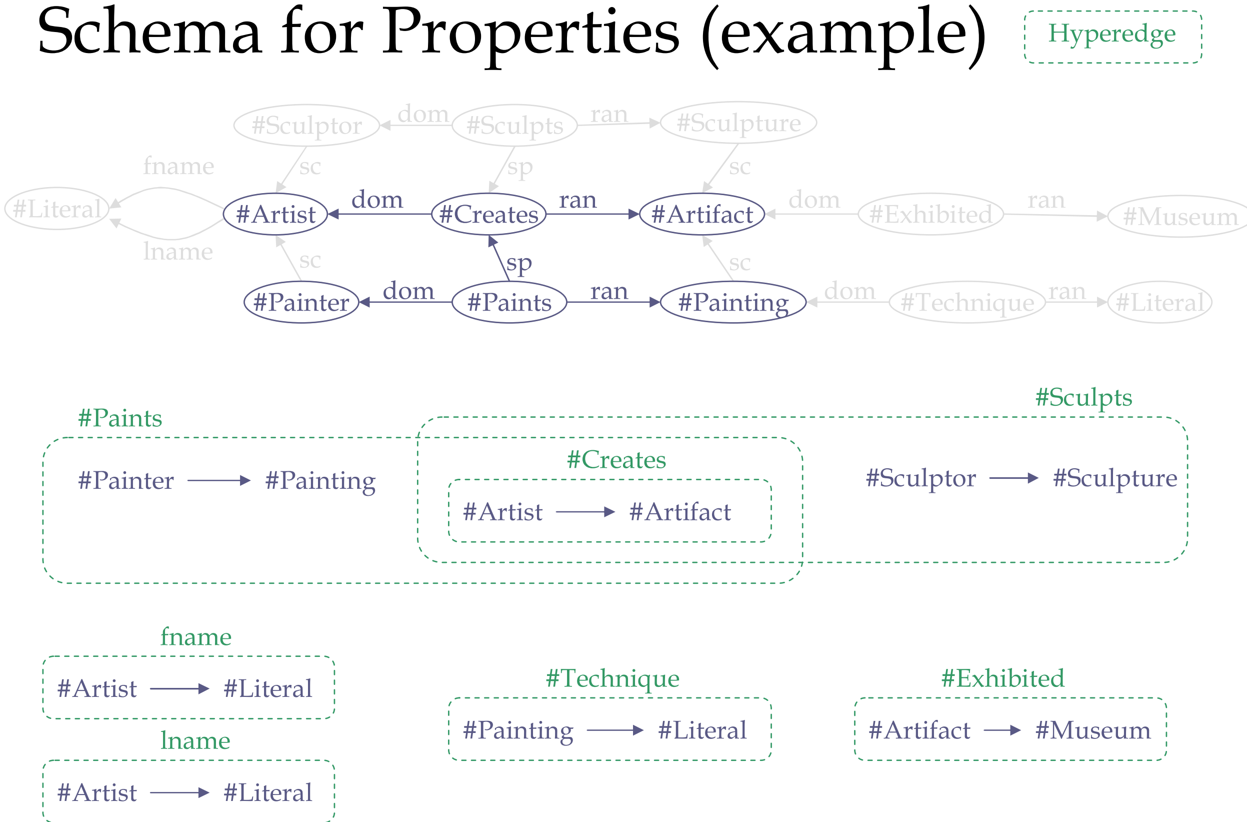


Schema for Properties (definition)

A *Property Graph Schema* is a graph schema (V, E_d, E_u) with the following interpretation:

- A hyperedge $(c, P) \in E_u$ represents a *Property Class* where:
  - c is the name of the class
  - Each edge $(u, v) \in P$ represents a valid pair *(domain,range)* for the property class
- If a property class c_1 contains a property class c_2, then c_1 is a *subPropertyOf* c_2 (i.e. inheritance of domains and ranges)

Schema for Properties (example)



Nested Graph (Hypernode)

A *Nested Graph (nGraph)* is defined recursively as a triple (n, V_n, E_n) where:

(a) $n \in U$ is the name of the nGraph

(b) V_n is a finite set of vertices, such that, each $v \in V_n$ satisfies that $v \in U \cup L$ or v is a nGraph

(c) $E_n \subseteq V_n \times V_n$ is a finite set of unlabeled directed edges, such that, each triple $(u, v) \in E_n$ satisfies that:
(i) $u \in U$ or u is a nGraph
(ii) $v \in U \cup L$ or v is a nGraph

Resource Graph Instance (definition)

A *Resource Graph Instance* is a pair (V_I, E_I), where:

(1) V_I is a finite set of nGraphs, and

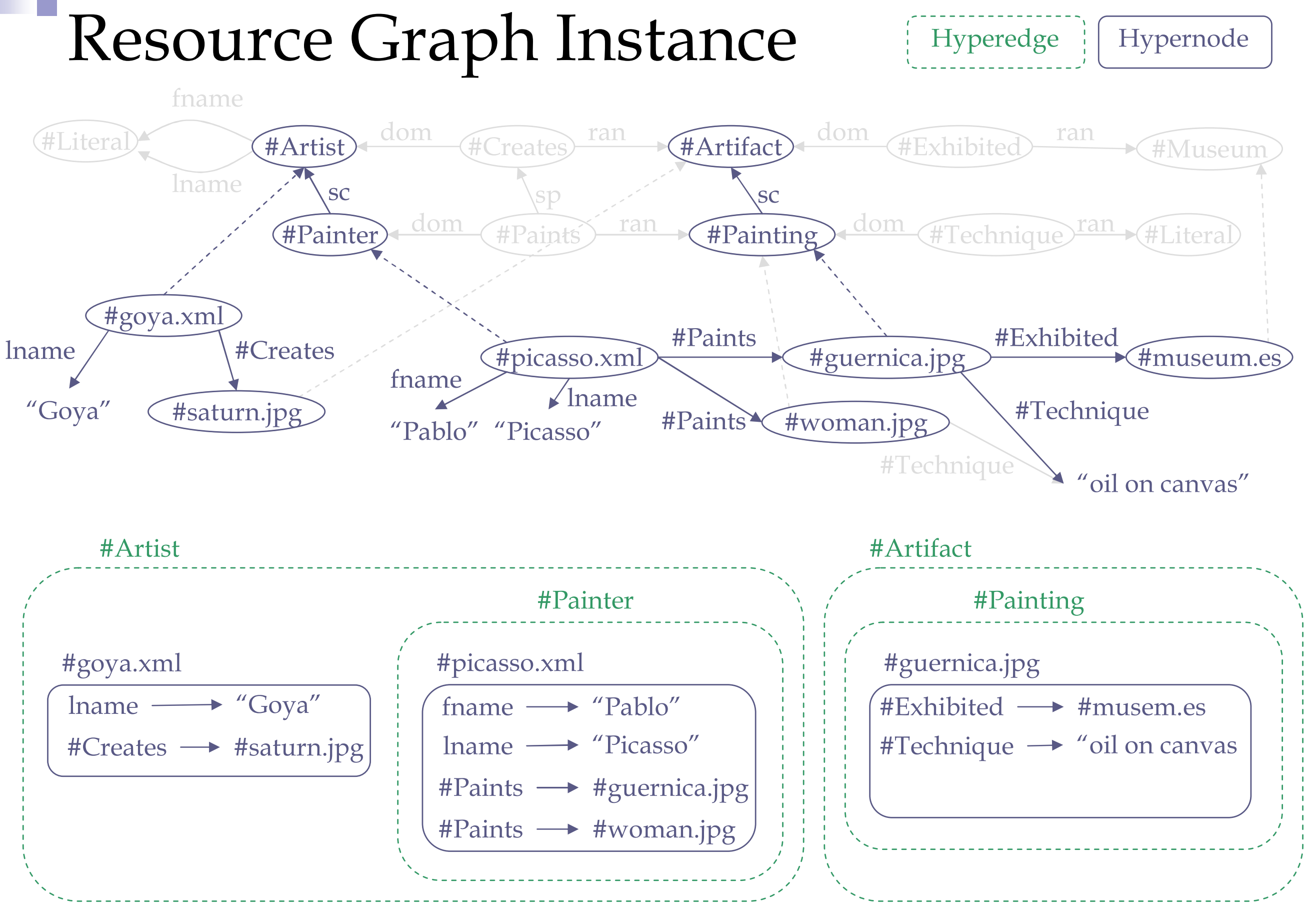
(2) E_I is a finite set of labeled undirected hyperedges (just hyperedges) of the form (c, R) satisfying that:
- $c \in U$ is a resource class name from the schema
- $R \subseteq V_I$ is the set of nGraphs of type c

Restrictions:

(C1) Given two nGraphs (n, V_n, E_n), $(n', V_{n'}, E_{n'}) \in V_I$, $n = n'$ implies that $V_n = V_{n'}$ and $E_n = E_{n'}$

(C2) $\forall (n, V_n, E_n) \in V_I$ it is not the case that $n \in V_{n'}$ for any $(n', V_{n'}, E_{n'}) \in V_I$

Resource Graph Instance



Querying RDF data

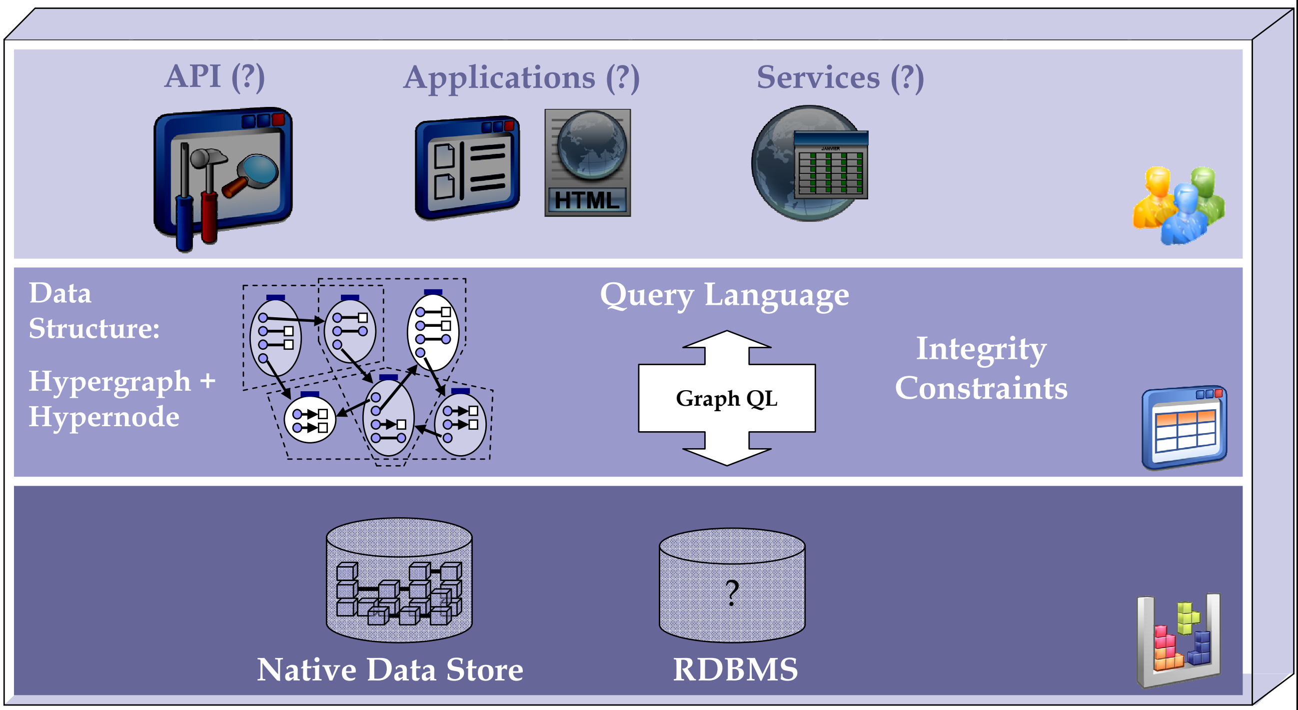
Graph notion	Real-life Application Query
Adjacency	**All relatives of degree one of Alice** **(**adjacent nodes in a genealogy database)
	What chemical composes does a given chemical reaction produce? (Adjacent edges in chemical information)
	What cities are near Athens? (neighborhood in a tourism system graph)
Degree of a node	**What is/are the most cited paper/s?** (searching node/s with maximum in-degree in a database of bibliographic cites)

Querying RDF data (cont)

Graph notion	Real-life Application Query
Paths	**Are suspects A and B related?** (relevant paths in a police database)
	What is the shortest route between city A and city B? (Shortest path in a database of roads)
	What is the influence of article D? (transitive closure in database of bibliographic cites)
Distance	**What is the Erdös distance between author X and author Y?** (distance between nodes in a collaboration graph)
Pattern Matching	**Where and how much a motif (pattern) appears?** (Pattern matching in genome data)

PROPERTY		Adjacent Nodes	Adjacent Edges	Degree of a Node	Path	Fixed-length path	Distance	Diameter
RDF Query Language	RQL	Yellow	Yellow	Yellow	Red	Yellow	Red	Red
	SeRQL	Yellow	Yellow	Red	Red	Yellow	Red	Red
	RDQL	Yellow	Yellow	Red	Red	Yellow	Red	Red
	Triple	Yellow	Yellow	Red	Red	Yellow	Red	Red
	N3	Yellow	Red	Red	Red	Yellow	Red	Red
	Versa	Yellow	Red	Red	Red	Red	Red	Red
	RxPath	Red	Red	Red	Yellow	Yellow	Red	Red
	Sparql	Yellow	Yellow	Yellow	Red	Yellow	Red	Red
Graph Query Language	G	Yellow	Yellow	Red	Green	Green	Red	Red
	G+	Green	Green	Green	Green	Green	Green	Green
	Graph Log	Green	Green	Green	Green	Green	Green	Green
	Gram	Green	Green	Red	Green	Green	Red	Red
	Graph DB	Yellow	Yellow	White	Green	Green	White	White
	Lorel	Green	Green	Red	Green	Green	Red	Red
	F-G	Yellow	Yellow	Red	Green	Green	Red	Red

Current and future work



Thanks

Questions?