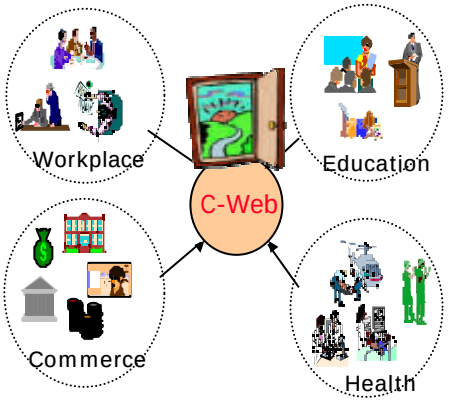
Querying RDF descriptions for Community Web Portals

G. Karvounarakis V. Christophides
D. Plexousakis S. Alexaki

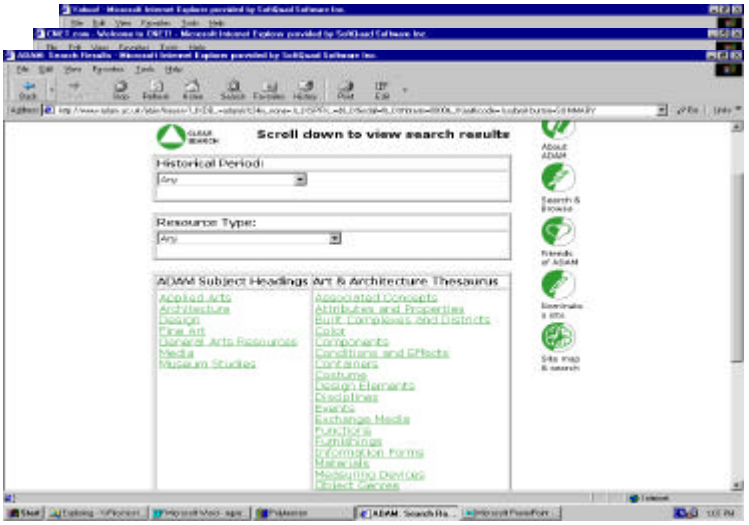
Computer Science Department, University of Crete
and
Institute for Computer Science - FORTH
Heraklion, Crete

What is a Community Web ?

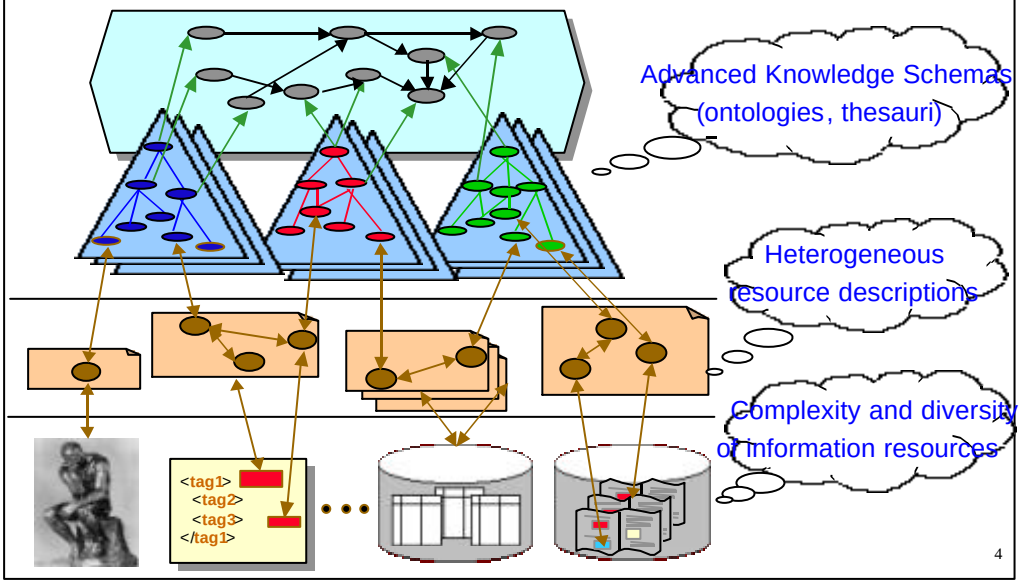
- A group of people sharing a domain of discourse and a set of information resources (e.g., data, documents, services) and having some common interests
  - Commerce, Education, Health
- The main requirement is to provide a single point of useful, ubiquitous comprehensive, and integrated access to various information resources
  - Web Portals



Portalmania!



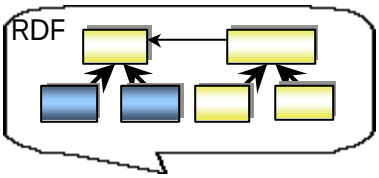
Community Web Portals: Main Features



What do we need?

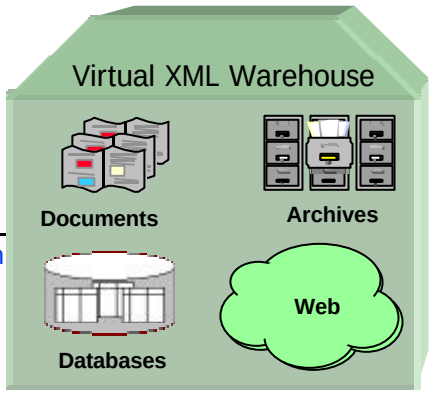
- Flexible Modeling of Community Web Portal Catalogs
  - ❶ Exploit existing forms of community knowledge
    - Ranging from simple vocabularies to formal ontologies
  - ❷ Describe in various ways community information resources
    - Administration, Classification, Copyrights, Content Rating, Channels, ….
- Secondary Storage Management of Portal Metadata
  - ❶ Large Schemas: e.g., 170 Mbytes of ODP Topics (the Art Hierarchy contains 25315 terms)
  - ❷ Voluminous Description Bases: e.g., 700 Mbytes of ODP indexed sites (2,342,978 URLs)
- Declarative Query Languages for Portal Catalogs
  - ❶ Interleave schema with data querying
  - ❷ Optimize access to Portal Catalogs

Our Approach





High-level Access to community information

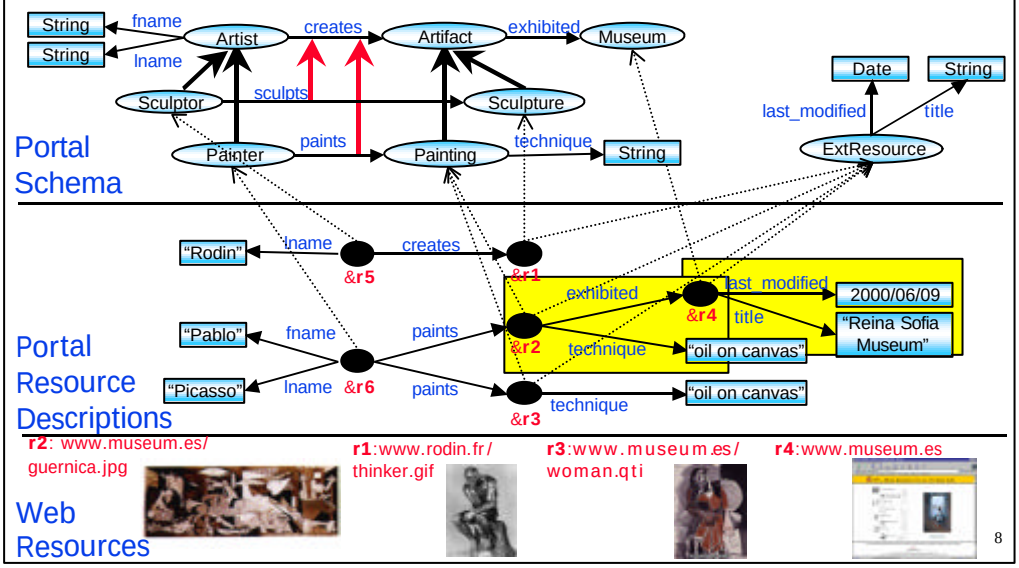


- Use W3C Standards to describe (RDF/S) & exchange (XML) information
- Our Main Contribution: Declarative Languages for Browsing & Querying

Outline

- Example of a Portal Catalog for Cultural Communities
- Describing and Querying Community Resources
  - A Formal Data Model for RDF/S
  - The RDF Query Language (RQL)
- RDF Suite Architecture
  - RQL Query processing over ORDBMS
- Summary

Building a Cultural Community Web Portal using RDF



Resource Description Framework (RDF/S)

- RDF: Resource Descriptions
  - Data Model: Directed Labeled Graphs
    - Nodes: Resources (URIs) or Literals
    - Edges: Properties – Attributes or Relationships
    - Labels: Nodes (Class names) and Edges (Property names)
    - Statement: assertion of the form *resource, property, value*
    - Description: collection of statements concerning a resource
  - XML syntax
- RDF Schema (RDFS): Schema Vocabularies
  - Specialization of both classes & properties (simple & multiple)
  - Multiple classification under several classes
  - Unordered, optional, and multi-valued properties
  - Domain and range polymorphism of properties

RDF/S vs. Well-Known Formalisms

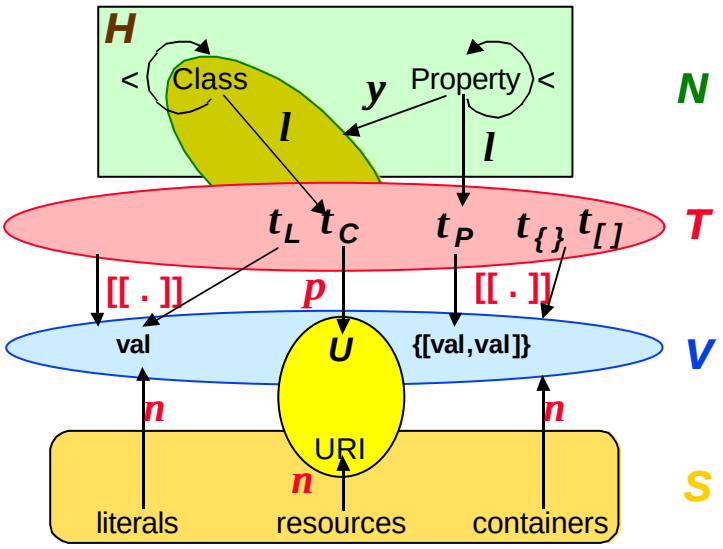
- Relational or Object Database Models (ODMG, SQL)
  - Classes don't define table or object types
  - Instances may have associated quite different properties
  - Collections with heterogeneous members
- Semistructured or XML Data Models (OEM, UnQL, YAT, XML Schema)
  - Labels only on nodes or edges
  - Class and property subsumption is not captured
  - Heterogeneous structures reminiscent to SGML exceptions
- Knowledge Representation Languages (Telos, DL, F-Logic)
  - Absence of complex values and n-ary relationships (bags, sequences)

A Formal Data Model for RDF

- *Ç(Í,<)* is a *well-defined* hierarchy of classes/properties names iff:
  - c ∈ **C** => c < Class
  - p ∈ **P** => p < Property
  - $p_1, p_2 \in$ **P** and $p_1 < p_2$ => *domain*(p_1) ≤ *domain*(p_2) and *range*(p_1) ≤ *range*(p_2)
- Type System:

  $t = t_L \mid t_U \mid \{t\} \mid [t] \mid (1{:}t + 2{:}t + \ldots + n{:}t)$
- Interpretation Function:
  - Literal types, $[[\ t_L\ ]] = dom(t_L)$
  - Bag types, $[[\ \{t\}\ ]] = \{v_1, v_2, \ldots, v_n\}$, $v_1, v_2, \ldots, v_n \in$ **V** are values of type t
  - Seq types, $[[\ [t]\ ]] = \{v_1, v_2, \ldots, v_n\}$, $v_1, v_2, \ldots, v_n \in$ **V** are values of type t
  - Alt types, $[[\ (1{:}t_1 + 2{:}t_2 + \ldots + n{:}t_n)\ ]] = v_i$, $v_i \in$ **V**, 1<i<n is a value of type t_i
  - c ∈ **C,** [[c]] = {v | v ∈ π(c)}∪{π(c') | c' < c}
  - p ∈ **P,** [[p]] = {[v_1, v_2] | v_1 ∈ [[*domain*(p)]], v_1 ∈ [[*range*(p)]]}∪{π(p') | p' < p}

A Formal Data Model for RDF



The RDF Query Language

- Declarative query language for RDF description bases
  - relies on a typed data model (literal & container types + union types)
  - follows a functional approach (basic queries and filters)
  - adapts the functionality of semistructured or XML query languages to RDF, but also:
    - treats properties as self-existent individuals
    - exploits taxonomies of node and edge labels
    - allows querying of schemas as semistructured data
- Relational interpretation of schemas & resource descriptions
  - Classes (unary relations)
  - Properties (binary relations)
  - Containers (n-ary relations)

Querying Large Portal Schemas with RQL

- Basic Class Queries
  - topclass
  - subclassof(Artist)
  - subclassof^(Artist)
  - superclassof(Painter)
  - superclassof^(Painter)
- Basic Property Queries
  - topproperty
  - subpropertyof(creates)
  - subpropertyof^(creates)
  - superpropertyof(paints)
  - superpropertyof^(paints)
- Querying the RDF/S meta-schema
  - Class
  - Property
  - Literal



Class & Property Querying

- Find the domain and range of the property creates

  seq (domain(creates), range(creates))

- Which classes can appear as domain and range of property creates

  select $X, $Y from {$X}creates{$Y} or

  select X, Y from Class{X}, Class{Y}, {:X}creates{:Y}

- Find all properties defined on class Painting and its superclasses

  select @P, range(@P) from {:Painting}@P or

  select P, range(P)
  from Property{P}
  where domain(P) >= Painting



RQL Query Result

property exhibited	class Museum
property technique	literal string

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  - <rdf:Bag ID="bag177976867">
    - <rdf:li>
      - <rdf:Seq>
          <rdf:li rdf:type="property" rdf:resource="exhibited" />
        - <rdf:li>
          - <rdf:Alt>
              <rdf:li rdf:type="class" rdf:resource="Museum" />
            </rdf:Alt>
          </rdf:li>
        </rdf:Seq>
      </rdf:li>
    - <rdf:li>
      - <rdf:Seq>
          <rdf:li rdf:type="property" rdf:resource="technique" />
        - <rdf:li>
          - <rdf:Alt>
              <rdf:li rdf:type="literal" rdf:resource="string" />
            </rdf:Alt>
          </rdf:li>
        </rdf:Seq>
      </rdf:li>
    </rdf:Bag>
</RDF>
```

Querying Complex Portal Descriptions with RQL

- Find the resources in the extent of the property creates

```
creates     or
select * from {x}creates{y}
```

Includes paints & sculpts

- Find the resources of type ExtResource and Sculpture

```
ExtResource intersect Sculpture
```

Multiply classified resources

- Find the most recently modified resources

```
select x , y
from {x}last_modified{y}
where y = max ( bag ( y ) )
```

Aggregate functions



Using Schema Information to Filter Data

- Navigational queries on semistructured resource descriptions
  - *Find the properties emanating from ExtResources and their source and target values*

```
select  x , @P , y
from    {x:ExtResource}@P{y}
```

Data paths foreseen in the schema

  - *Find the properties applied on instances of the class ExtResource and their source and target values*

```
select  x, @P, y
from    ExtResource{x}.@P{y}
```

Data paths not foreseen in the schema



Notice the difference

resource	property	resource
http://www.museum.es/guernica.jpg	exhibited	http://www.museum.es
resource	property	string
http://www.museum.es/guernica.jpg	technique	oil on canvas
resource	property	string
http://www.museum.es/woman.qti	technique	oil on canvas
resource	property	string
http://www.museum.es	title	Reina Sofia Museum
resource	property	date
http://www.museum.es	last_modified	2000-06-09T12:30:34+00:00

Putting it all Together

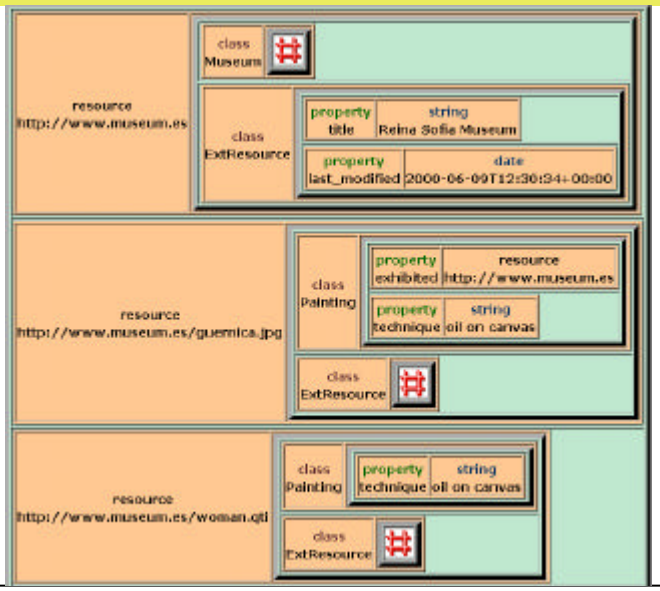
- Discover Schema and Data of resource descriptions
  - *Find the descriptions of resources whose URI match "www.museum.es"*

Implicit join condition on resource and class nodes

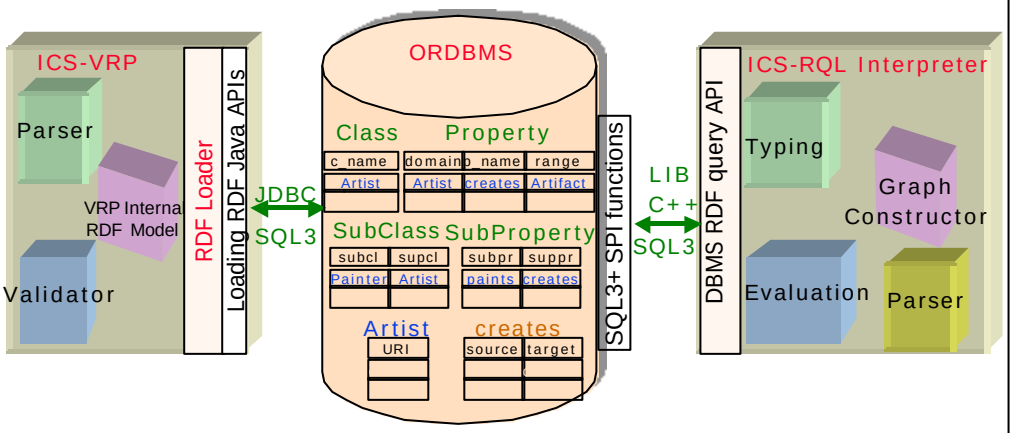
```
select X, (select $W, (select @P, Y
                       from {X : $W} @P {Y})
           from $W {X})
from Resource {X}
where X like "www.museum.es"
```



RQL Query Result



The RDFSuite Architecture



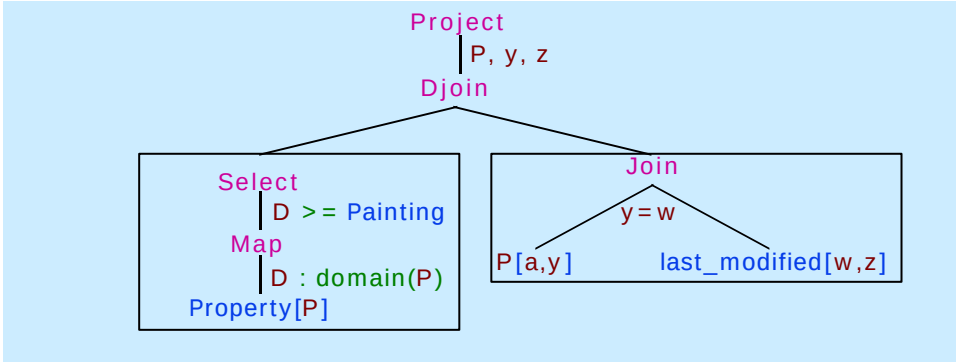
RQL Query Processing

- *Find the resources with a last_modified property, that can be reached by a property applied to the class Painting and its subclasses*

```
select R, y , z
from {:Painting}@P, @P{y}.last_modified{z}
```

property	resource
exhibited	http://www.museum.es

RQL Query Execution Plans



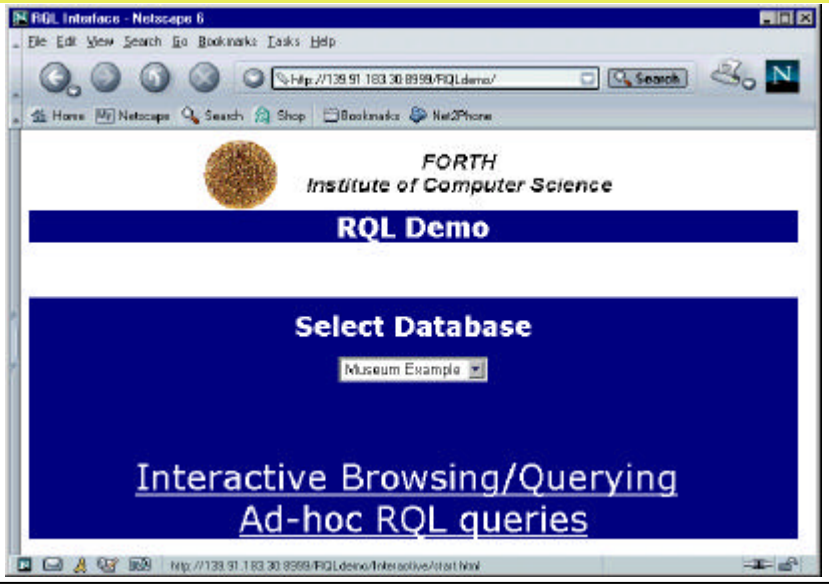

```
select P
from   Property P
where issubclassof(Painting, domain(P))
```

```
select Q.target, L.target
from   P Q, last_modified L
where Q.target = L.source
```

Summary

- Our contribution:
  - First declarative language for uniformly querying RDF schemas and resource descriptions
  - Formal model for RDF/S (+ typed, operational semantics)
- Functionality evaluation:
  - Developed in the context of the EC project C-Web (http://cweb.inria.fr)
  - Accepted for use in the ongoing EC projects:
    - MesMuses (http://aquarelle.inria.fr/mesmuses)
    - Ontoknowledge (http://www.ontoknowledge.org)
    - Arion (http://dlforum.external.forth.gr:8080)
- Performance evaluation:
  - Testbed: Open Directory RDF dump
  - Optimization opportunities:
    - Schema and Data Indexing Techniques (transitive closure queries)
    - Heuristics for Algebraic Transformations (schema and data queries)

http://139.91.183.30:8999/RQLdemo/



Comparing RQL to W3C XQuery

- Find the names of those who have created artifacts which are exhibited in Museums, along with the Museum titles
  - RQL

```
select Y, Z, V, R
from {X}creates.exhibited{Y}.title{Z},
     {X}first_name{V},{X}last_name{R}
```


Comparing RQL to W3C XQuery

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  - <rdf:Bag ID="bag2287967">
    - <rdf:li>
      - <rdf:Seq>
          <rdf:li rdf:type="resource" rdf:resource="http://www.rodin.fr" />
          <rdf:li rdf:type="string">Rodin Museum</rdf:li>
          <rdf:li rdf:type="string">August</rdf:li>
          <rdf:li rdf:type="string">Rodin</rdf:li>
        </rdf:Seq>
      </rdf:li>
    - <rdf:li>
      - <rdf:Seq>
          <rdf:li rdf:type="resource" rdf:resource="http://www.museum.es" />
          <rdf:li rdf:type="string">Reina Sofia Museum</rdf:li>
          <rdf:li rdf:type="string">Pablo</rdf:li>
          <rdf:li rdf:type="string">Picasso</rdf:li>
        </rdf:Seq>
      </rdf:li>
    - <rdf:li>
      - <rdf:Seq>
          <rdf:li rdf:type="resource" rdf:resource="http://www.louvre.fr/" />
          <rdf:li rdf:type="string">Louvre Museum</rdf:li>
          <rdf:li rdf:type="string">Michelangelo</rdf:li>
          <rdf:li rdf:type="string">Buonarroti</rdf:li>
        </rdf:Seq>
      </rdf:li>
    </rdf:Bag>
</RDF>
```

Comparing RQL to W3C XQuery

- Xquery

```
LET  $t := document("sirpac-culture-merged.rdf")//description
FOR $artist IN rdf:instance-of-class($t, rdf:predicate-domain($t,
  "creates"))
LET $artifact := rdf:join-on-property($t, $artist,"creates"),
    $museum := rdf:join-on-property($t, $artifact, "exhibited")
RETURN
    <result>
      {filter($artist | $artist/last_name | $artist/first_name),
       filter($museum | $museum/title)}
    </result>
```


Comparing RQL to W3C XQuery

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  - <result>
    - <description rdf:about="rodin424">
        <first_name>August</first_name>
        <last_name>Rodin</last_name>
      </description>
    - <description rdf:about="http://www.rodin.fr">
        <title>Rodin Museum</title>
      </description>
    </result>
  - <result>
    - <description rdf:about="michelangelo">
        <first_name>Michelangelo</first_name>
        <last_name>Buonarroti</last_name>
      </description>
    - <description rdf:about="http://www.louvre.fr/">
        <title>Louvre Museum</title>
      </description>
    </result>
  - <result>
    - <description rdf:about="picasso132">
        <first_name>Pablo</first_name>
        <last_name>Picasso</last_name>
      </description>
    - <description rdf:about="http://www.museum.es">
        <title>Reina Sofia Museum</title>
      </description>
    </result>
</RDF>
```

Comparing RQL to W3C XQuery

- XML syntactic and schematic discrepancies of semantically equivalent RDF statements
  - normalized representation under the form of merged descriptions
- XQuery has no built-in knowledge of the RDF schema information
  - function library that exploits the RDF schema if the assertions of the schema are also present in the normalized representation
- Data model mismatches between XML and RDF impact type safety of functions and queries
  - bag (range(Artist)) union subclassof (Artifact)
    - In RQL Type Error
    - In XQuery All the subclasses of Artifact

31