Big Graph Processing Systems

Part III: RDF and SPARQL

▶ **Chapter 1:** RDF in a Nutshell

Christopher Spinrath

CNRS – LIRIS – Lyon 1 Université

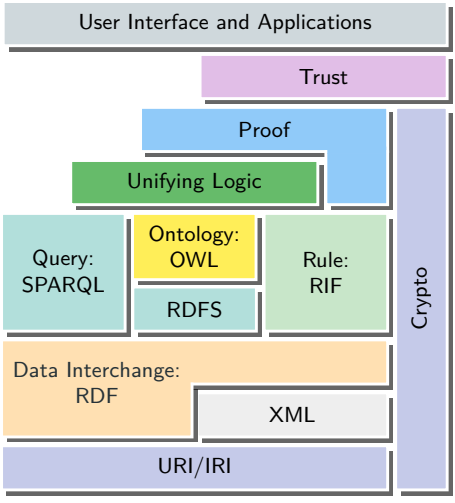
DISS Master 2025

This presentation is based on slides by E. Della Valle

`http://emanueledellavalle.org` – @manudellavalle

A Very Brief History on RDF and SPARQL

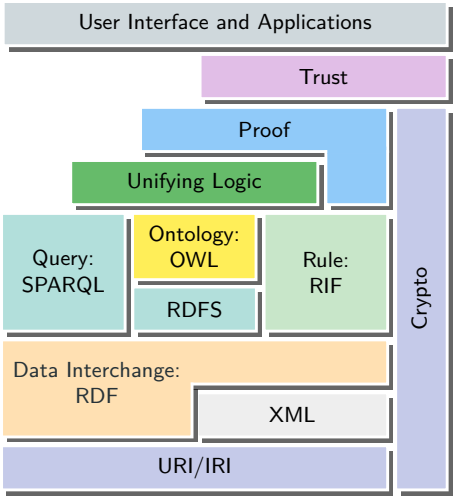
- RDF was published as a W3C recommendation in 1999
- Originally intended for describing metadata
- Part of the Semantic Web Initiative
- Later generalized to cover all kinds of data/knowledge



A Very Brief History on RDF and SPARQL

- RDF was published as a W3C recommendation in 1999
- Originally intended for describing metadata
- Part of the Semantic Web Initiative
- Later generalized to cover all kinds of data/knowledge

- Data model of RDF Triplestores – a type of graph database



RDF – The Resource Description Framework

RDF is short for Resource Description Framework

Resource: everything that can have a unique identifier, for instance web pages, places, people, products, etc.

Description: attributes, features, and relations of the resources

Framework: model, languages, syntax (serializations) for the descriptions

RDF – The Resource Description Framework

RDF is short for Resource Description Framework

Resource: everything that can have a unique identifier, for instance web pages, places, people, products, etc.

Description: attributes, features, and relations of the resources

Framework: model, languages, syntax (serializations) for the descriptions

- Very flexible data model
- Due to its origins in the semantic web initiative design goals covered
  - data being distributed ...
  - ...without a central authority, and
  - the graceful evolution of data (data and applications are always changing)

RDF Triples

In RDF data is described by means of triples consisting of

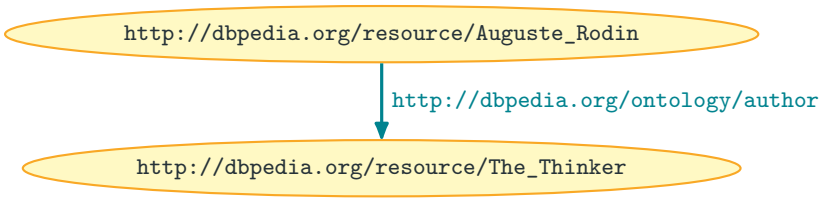
- a subject – representing a resource,
- a predicate (or property) – representing a relationship, and
- an object (or value) – representing a resource or a literal

RDF Triples

In RDF data is described by means of triples consisting of

- a subject – representing a resource,
- a predicate (or property) – representing a relationship, and
- an object (or value) – representing a resource or a literal

Example



Internationalized Resource Identifiers

Two Important Problems

of distributed databases, w/o central authority

1. How to assign (unique) IDs?
2. How to name schema elements?
   (e.g. predicates)

Internationalized Resource Identifiers

Two Important Problems

of distributed databases, w/o central authority

1. How to assign (unique) IDs?
2. How to name schema elements? (e.g. predicates)

Internationalized Resource Identifier (IRI)

- RDF uses IRIs to address the above problems
- IDs and schema names are IRIs
- Values do not (always) need to be IRI-fied

Internationalized Resource Identifiers

Two Important Problems

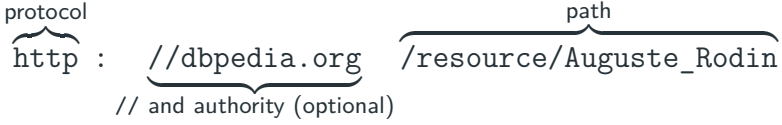
of distributed databases, w/o central authority

1. How to assign (unique) IDs?
2. How to name schema elements? (e.g. predicates)

Internationalized Resource Identifier (IRI)

- RDF uses IRIs to address the above problems
- IDs and schema names are IRIs
- Values do not (always) need to be IRI-fied

Example (IRI)



In general there are two more optional components: query (?...) and fragment (#...)

Internationalized Resource Identifiers

Two Important Problems

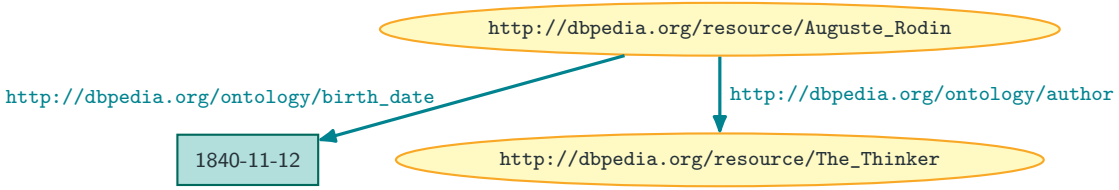
of distributed databases, w/o central authority

1. How to assign (unique) IDs?
2. How to name schema elements? (e.g. predicates)

Internationalized Resource Identifier (IRI)

- RDF uses IRIs to address the above problems
- IDs and schema names are IRIs
- Values do not (always) need to be IRI-fied

Example (RDF Graph)



Internationalized Resource Identifiers

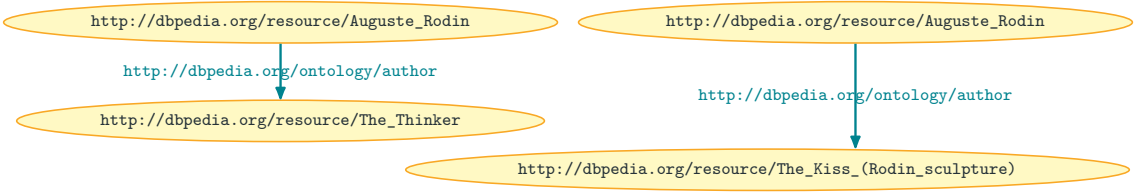
Q: Which IRIs should we use?
A: Popular ones! Data merge will take place automatically!

Internationalized Resource Identifiers

Q: Which IRIs should we use?

A: Popular ones! Data merge will take place automatically!

Example

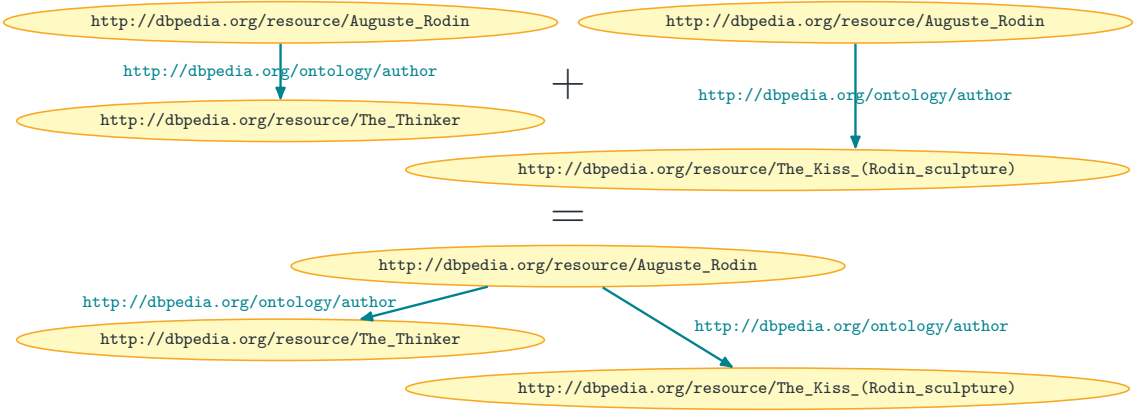


Internationalized Resource Identifiers

Q: Which IRIs should we use?

A: Popular ones! Data merge will take place automatically!

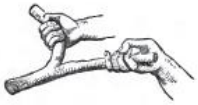
Example



Internationalized Resource Identifiers

Where Do I Find Popular IRIs?

1. Ontology Dowsing: `http://www.w3.org/wiki/Ontology_Dowsing`



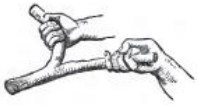
Internationalized Resource Identifiers

Where Do I Find Popular IRIs?

1. Ontology Dowsing: `http://www.w3.org/wiki/Ontology_Dowsing`
2. Use well know ones ;-)
   - RDF vocabulary already includes some essential base types
     - `rdf:type`: the "instance of" a relationship between an individual and its class
     - `rdf:Property`: the "concept" of a property

Internationalized Resource Identifiers

Where Do I Find Popular IRIs?



1. Ontology Dowsing: `http://www.w3.org/wiki/Ontology_Dowsing`
2. Use well know ones ;-)
   - RDF vocabulary already includes some essential base types
     - `rdf:type`: the "instance of" a relationship between an individual and its class
     - `rdf:Property`: the "concept" of a property
   - RDF comes along a schema description language (RDF-S)
     - `rdfs:Class`: the "concept" of a class
     - `rdfs:subClassOf`: the "is a" relationship between two classes
     - `rdfs:label`: the standard way to provide a human-readable label
     - etc.

Internationalized Resource Identifiers

Where Do I Find Popular IRIs?

1. Ontology Dowsing: `http://www.w3.org/wiki/Ontology_Dowsing`
2. Use well know ones ;-)
   - RDF vocabulary already includes some essential base types
     - `rdf:type`: the "instance of" a relationship between an individual and its class
     - `rdf:Property`: the "concept" of a property
   - RDF comes along a schema description language (RDF-S)
     - `rdfs:Class`: the "concept" of a class
     - `rdfs:subClassOf`: the "is a" relationship between two classes
     - `rdfs:label`: the standard way to provide a human-readable label
     - etc.
3. Search for them
   - `https://schema.org/` is a community maintained repository
   - `https://interoperable-europe.ec.europa.eu/` is a repository from the European commission
   - etc.

Internationalized Resource Identifiers

Person

A Schema.org Type

Thing › Person

[more...]

A person (alive, dead, undead, or fictional).

Property	Expected Type	Description
Properties from **Person**		
additionalName	Text	An additional name for a Person, can be used for a middle name.
address	PostalAddress or Text	Physical address of the item.
affiliation	Organization	An organization that this person is affiliated with. For example, a school/university, a club, or a team.
agentInteractionStatistic	InteractionCounter	The number of completed interactions for this entity, in a particular role (the 'agent'), in a particular action (indicated in the statistic), and in a particular context (i.e. interactionService).
alumniOf	EducationalOrganization or Organization	An organization that the person is an alumni of. Inverse property: alumni

[https://schema.org/Person, accessed 2025-02-14]

Internationalized Resource Identifiers

What is a value? When shall we IRI-fy a value?

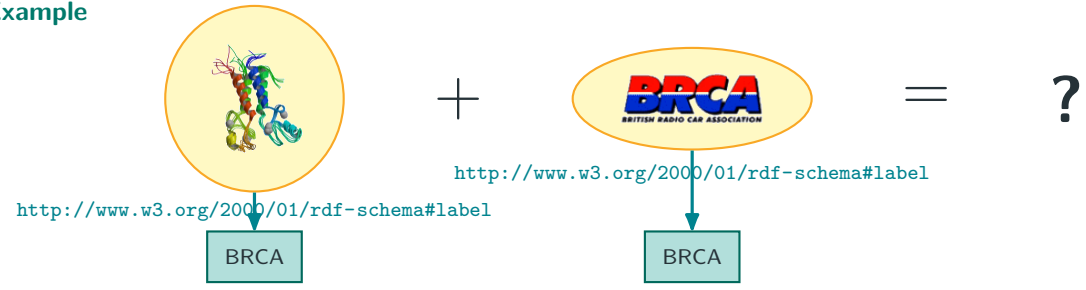
- For instance, what if we want to merge two resources based on their labels?
- Literals cannot be used to merge different data sets

Internationalized Resource Identifiers

What is a value? When shall we IRI-fy a value?

- For instance, what if we want to merge two resources based on their labels?
- Literals cannot be used to merge different data sets

Example



- BRCA may refer to different thing on the Web (see `https://en.wikipedia.org/wiki/BRCA`)

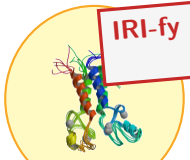
Internationalized Resource Identifiers

What is a value? When shall we IRI-fy a value?

- For instance, what if we want to merge two resources based on their labels?
- Literals cannot be used to merge different data sets

Example

IRI-fy any value that can eventually be used to merge different dataset and leave the other values as literals



http://www.w3.org/2000/01/rdf-schema#label

BRCA

+

http://www.w3.org/2000/01/rdf-schema#label

BRCA

= ?

- BRCA may refer to different thing on the Web (see `https://en.wikipedia.org/wiki/BRCA`)

Data Integration and Interchange with RDF

RDF vs. the Relational Model

- The relational model is remarkably flexible and widely used

Q: Can relational data be represented with RDF triples?

RDF vs. the Relational Model

- The relational model is remarkably flexible and widely used

Q: Can relational data be represented with RDF triples? Yes!

From tuples to triples

Primary/Foreign Keys become subjects

Columns/Attributes become predicates

Values become literals

RDF vs. the Relational Model

- The relational model is remarkably flexible and widely used

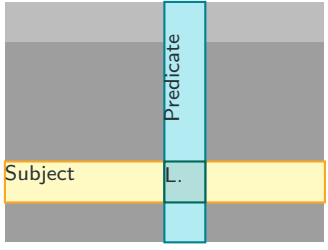
Q: Can relational data be represented with RDF triples? Yes!

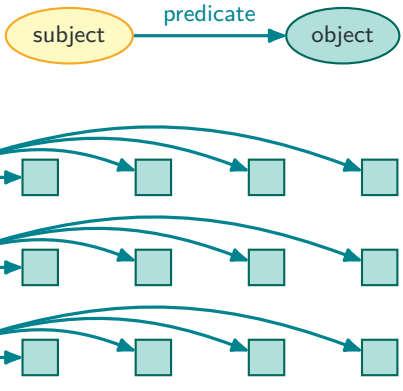
From tuples to triples

Primary/Foreign Keys become subjects

Columns/Attributes become predicates

Values become literals





RDF vs. the Relational Model

Example (Artist Data)

Sculpture

ID	Creator	Date
BB.2	Auguste Rodin	1889

Artwork Names

ID	Name	Lang
BB.2	The Kiss	en
BB.2	Il bacio	it
BB.2	Le Baiser	fr

RDF vs. the Relational Model

Example (Artist Data)

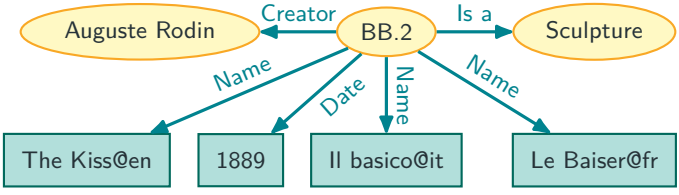
Sculpture

ID	Creator	Date
BB.2	Auguste Rodin	1889

Artwork Names

ID	Name	Lang
BB.2	The Kiss	en
BB.2	Il bacio	it
BB.2	Le Baiser	fr

Represented in RDF (almost)



RDF vs. the Relational Model

Example (Artist Data)

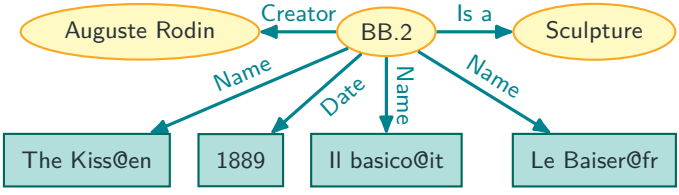
Sculpture

ID	Creator	Date
BB.2	Auguste Rodin	1889

Artwork Names

ID	Name	Lang
BB.2	The Kiss	en
BB.2	Il bacio	it
BB.2	Le Baiser	fr

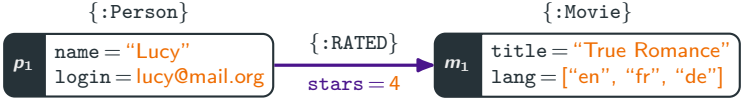
Represented in RDF (almost)



- IDs/Predicates are not IRI-fied
- Language tags @en, @it, @fr

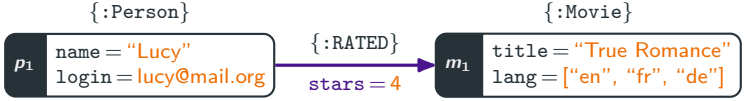
RDF vs. Property Graphs

Property Graph

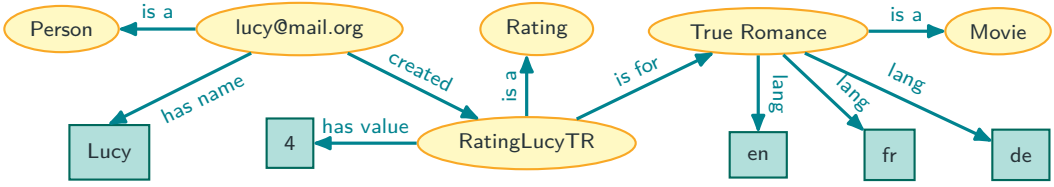


RDF vs. Property Graphs

Property Graph



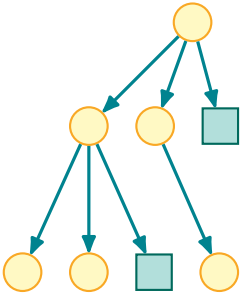
RDF Representation (almost)



- For readability, no IRIs

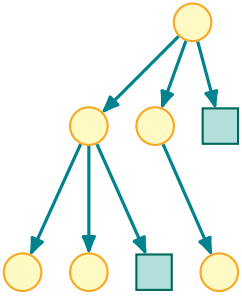
Representing Other Data Structures in RDF

Trees Can be Represented in RDF

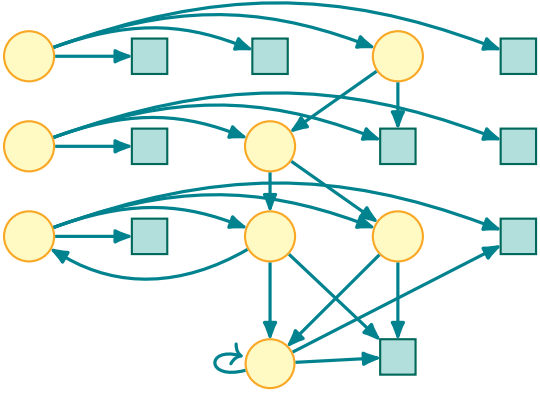


Representing Other Data Structures in RDF

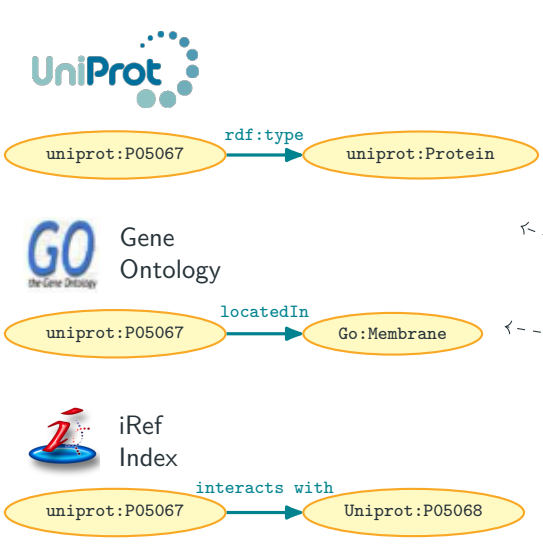
Trees Can be Represented in RDF



Anything Can be Represented in RDF



Multi-Source Data Integration with RDF



Serializing RDF

Serializing RDF

Serialization

Serialization is the process of translating a data structure or an object into a format that can be stored or transmitted

Serializing RDF

Serialization

Serialization is the process of translating a data structure or an object into a format that can be stored or transmitted

How to Serialize RDF Triples?

Many alternatives to write triples:

- XML, NTriples (CSV), Turtle, RDFa, JSON-LD, ...

Serializing RDF

Serialization

Serialization is the process of translating a data structure or an object into a format that can be stored or transmitted

How to Serialize RDF Triples?

Many alternatives to write triples:

- XML, NTriples (CSV), Turtle, RDFa, JSON-LD, ...

In this Lecture: Turtle

- Developer-friendly serialization
- Often used to import/export RDF Triplestore databases
- For example, `http://dbpedia.org/data/Propranolol.n3`

Serializing RDF

Serialization

Serialization is the process of translating a data structure or an object into a format that can be stored or transmitted

How to Serialize RDF Triples?

Many alternatives to write triples:

- XML, NTriples (CSV), Turtle, RDFa, JSON-LD, ...

In this Lecture: Turtle

- Developer-friendly serialization
- Often used to import/export RDF Triplestore databases
- For example, `http://dbpedia.org/data/Propranolol.n3`

Turtle Base Format

```
:subject1 :predicate1 "value1" .
:subject2 :predicate2 :subject1 .
```

Turtle – Namespaces

Example (Without Namespaces)

```
<http://dbpedia.org/resource/The_Kiss_(Rodin_sculpture)>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/Artwork> .
```

Turtle – Namespaces

Example (Without Namespaces)

```
<http://dbpedia.org/resource/The_Kiss_(Rodin_sculpture)>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/Artwork> .
```

IRIs can be Abbreviated using Namespaces

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sc: <http://schema.org/>

dbr:The_Kiss_(Rodin_sculpture) rdf:type dbo:Artwork .
```

Turtle – Language Tags and Types

Literals with Language Tags

Language tags can be written with an "@": `"Le Baiser"@fr`

```
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "The Kiss"@en .
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "Il Bacio"@it .
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "Le Baiser"@fr .
```

Turtle – Language Tags and Types

Literals with Language Tags

Language tags can be written with an "@": `"Le Baiser"@fr`

```
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "The Kiss"@en .
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "Il Bacio"@it .
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "Le Baiser"@fr .
```

Typed Literals

Types can be written with a double "^": `"1889"^^xsd:gYear`

```
dbr:The_Kiss_(Rodin_sculpture) sc:dateCreated "1889"^^xsd:gYear .
```

Turtle – Repeated Subjects

Example

```
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "The␣Kiss"@en .
dbr:The_Kiss_(Rodin_sculpture) sc:dateCreated "1889"^^xsd:gYear .
```

Turtle – Repeated Subjects

Example

```
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "The␣Kiss"@en .
dbr:The_Kiss_(Rodin_sculpture) sc:dateCreated "1889"^^xsd:gYear .
```

Abbreviating Repeated Subjects

- Using Semicola ";"

```
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "The Kiss"@en ;
                               sc:dateCreated "1889"^^xsd:gYear .
```

Turtle – Repeated Subjects Continued

Example

```
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "The Kiss"@en .
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "Il Bacio"@it .
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "Le Baiser"@fr .
```

Turtle – Repeated Subjects Continued

Example

```
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "The Kiss"@en .
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "Il Bacio"@it .
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "Le Baiser"@fr .
```

Abbreviating Repeated Subject/Predicate Pairs

- Using Commata ","

```
dbr:The_Kiss_(Rodin_sculpture) rdfs:label "The Kiss"@en,
                                          "Il Bacio"@it, "Le Baiser"@fr .
```

Turtle – Abbreviating Predicates

Example

```
dbr:The_Kiss_(Rodin_sculpture)
    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> dbo:Artwork .
```

Turtle – Abbreviating Predicates

Example

```
dbr:The_Kiss_(Rodin_sculpture)
    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> dbo:Artwork .
```

Abbreviating Predicates

- Using the assignment operator "="

```
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type> = 'a'

dbr:The_Kiss_(Rodin_sculpture) a dbo:Artwork .
```

Blank Nodes

Blank Nodes

Q: What if we cannot think about a good IRI?

A: When there is no good IRI, we can use blank nodes: 

Blank Nodes

Q: What if we cannot think about a good IRI?

A: When there is no good IRI, we can use blank nodes: 

Example

Suppose we want to represented the following relational table in RDF

Person	Bio Event	Date
Sofia	Birth	1974-02-28
Sofia	Marriage	1995-08-04

Blank Nodes

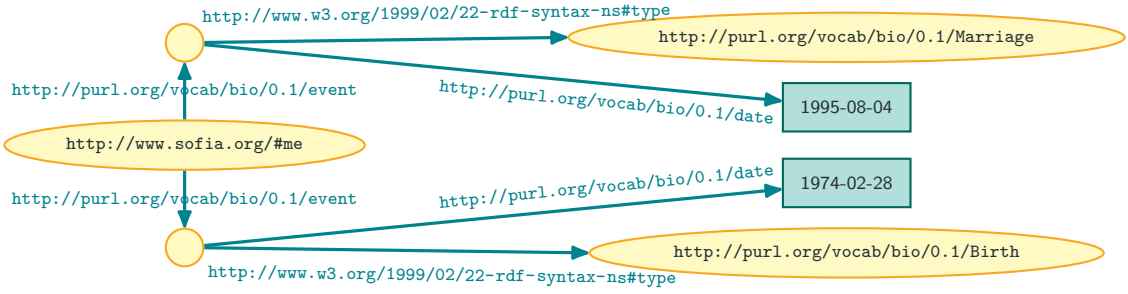
Q: What if we cannot think about a good IRI?

A: When there is no good IRI, we can use blank nodes:

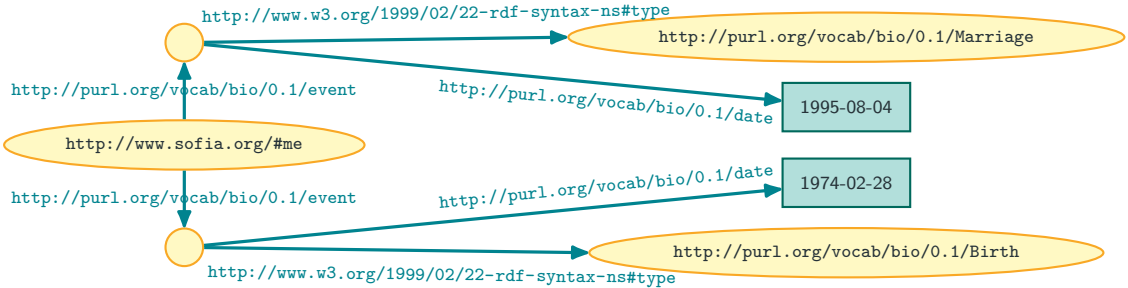
Example

Suppose we want to represented the following relational table in RDF

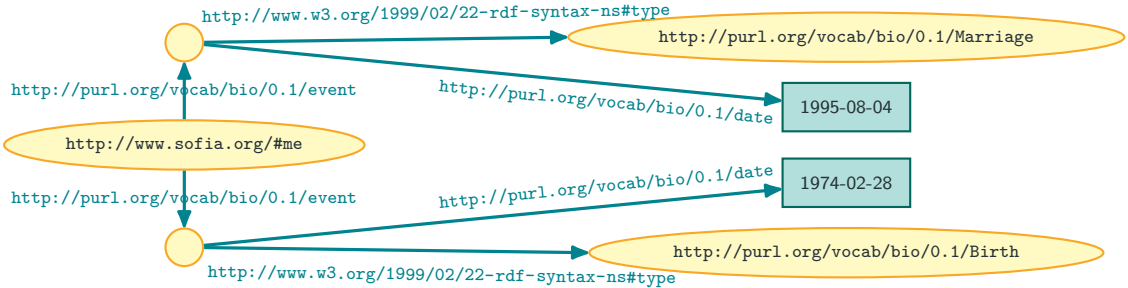
Person	Bio Event	Date
Sofia	Birth	1974-02-28
Sofia	Marriage	1995-08-04



Turtle – Serializing Blank Nodes



Turtle – Serializing Blank Nodes



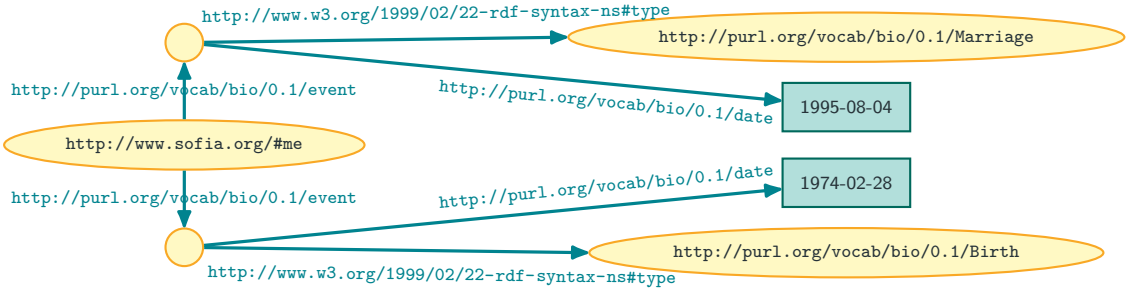
Serializing Blank Nodes

- Using blank node identifiers _:1, _:2, ...

```
@prefix bio: <http://purl.org/vocab/bio/0.1/> .

http://www.sofia.org/#me bio:event _:1, _:2 .
_:1 a bio:Birth; bio:date "1974-02-28"^^xsd:date .
_:2 a bio:Mariage; bio:date "1995-08-04"^^xsd:date .
```

Turtle – Serializing Blank Nodes



Serializing Blank Nodes

- Or square brackets […]

```
@prefix bio: <http://purl.org/vocab/bio/0.1/> .

http://www.sofia.org/#me bio:event
  [ a bio:Birth; bio:date "1974-02-28"^^xsd:date ],
  [ a bio:Mariage; bio:date "1995-08-04"^^xsd:date ].
```

Alternative RDF Serializations

Alternative RDF Serializations

1. RDF/XML: Original W3C standard based on XML

```
<rdf:Description rdf:about="http://...subject1">
  <predicate1>value</predicate1>
</rdf:Description>
```

Real-World example

Validator

Alternative RDF Serializations

1. RDF/XML: Original W3C standard based on XML

```
<rdf:Description rdf:about="http://...subject1">
  <predicate1>value</predicate1>
</rdf:Description>
```

Real-World example

Validator

2. NTriples: Simple (verbose) reference serialization
(for specifications only, Turtle without convenience syntax)

```
<http://...subject1> <http://...predicate1> "value1" .
<http://...subject2> <http://...predicate2> "value2" .
```

Alternative RDF Serializations

1. RDF/XML: Original W3C standard based on XML

```
<rdf:Description rdf:about="http://...subject1">
  <predicate1>value</predicate1>
</rdf:Description>
```

Real-World example

Validator

2. NTriples: Simple (verbose) reference serialization
(for specifications only, Turtle without convenience syntax)

```
<http://...subject1> <http://...predicate1> "value1" .
<http://...subject2> <http://...predicate2> "value2" .
```

3. Notation3 (N3): Superset of Turtle
(goes beyond RDF, can be used to describe rule)

Alternative RDF Serializations Cont'd

4. JSON-LD: Popular extension of JSON for RDF data

```
{
  "@context": "https://json-ld.org/contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John␣Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"
}
```

https://json-ld.org/

Alternative RDF Serializations Cont'd

4. JSON-LD: Popular extension of JSON for RDF data

```
{
  "@context": "https://json-ld.org/contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John␣Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"
}
```

https://json-ld.org/

5. RDFa: For the semantic web, embedding of RDF data in HTML via attributes

```
<div prefix="ov:http://open.vocab.org/terms/">
  <p vocab="http://www.sofia.org" resource="#me" typeof="Person">
    My name is <span property="name">Sofia</span>
  </p>
</div>
```

Credits

This presentation is adapted from and contains slides by

E. Della Valle – http://emanueledellavalle.org – @manudellavalle

It contains slides adapted from

- www.dcs.warwick.ac.uk/~acristea/courses/CS411/2010/SPARQL.ppt, and
- http://www.emse.fr/~zimmermann/Teaching/WebSem/SlidesSPARQL.html
- https://interoperable-europe.ec.europa.eu/sites/default/files/document/2014-02/D3.1.7%20Training%20Module%201.3%20Introduction%20to%20RDF_SPARQL_EUI_v0.08.pptx
