

# *Präsentation Metadaten*

Open Data Support wird von der Europäischen Kommission finanziert, gemäß SMART 2012/0107 'Lot 2: Provision of services for the Publication, Access and Reuse of Open Public Data across the European Union, through existing open data portals' (Vertrag No. 30-CE-0530965/00-17).

© 2013 European Commission



## OPEN DATA SUPPORT

Trainingsmodul 1.3

# Einführung in RDF & SPARQL

# Lernziele

Am Ende dieses Trainingsmoduls sollten Sie Folgendes kennen:

- Das Resource Description Framework (RDF/Quellenbeschreibungsgerüst);
- Wie schreibt/liest man RDF?
- Wie können Sie Ihre Daten mit RDF beschreiben?
- Was ist SPARQL?
- Welche verschiedenen Arten von SPARQL gibt es?
- Wie schreibt man eine SPARQL Abfrage?

# Inhalt

Dieses Modul enthält...

- Eine Einführung in das Resource Description Framework (RDF), um ihre Daten zu beschreiben.
  - Was ist RDF?
  - Wie ist es strukturiert?
  - Wie können Sie Ihre Daten in RDF darstellen?
- Eine Einführung in SPARQL, und wie können Sie es in RDF abfragen und bearbeiten.
- Hinweise zu weiterführender Literatur, Beispielen und Übungen.

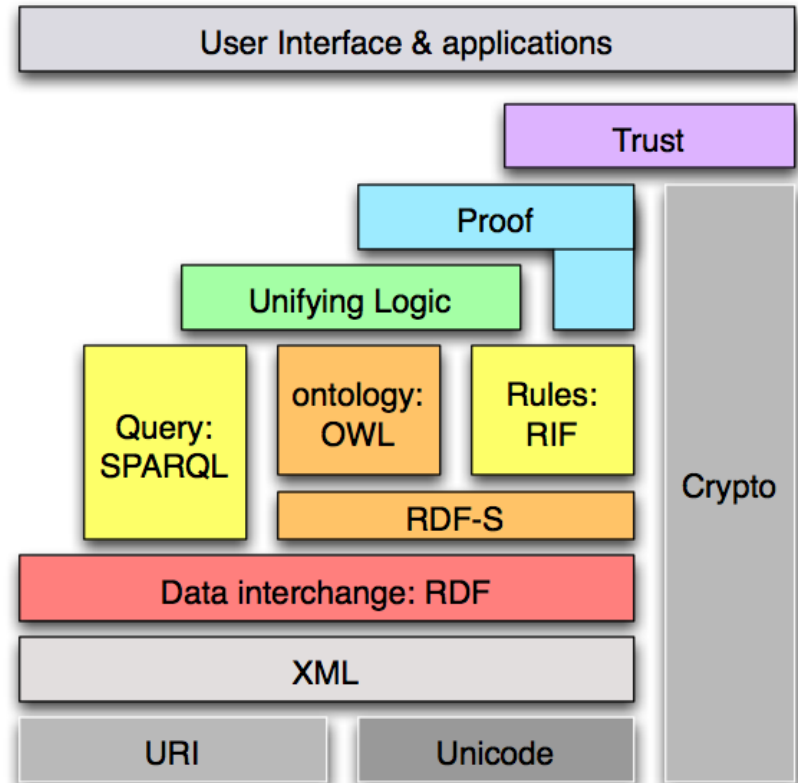
# Resource Description Framework

*Eine Einführung in RDF.*

# RDF in der Reihe von Semantic Web Technologien

RDF steht für:

- **Ressource (Quelle):** Alles, was einen einzigartigen Identifier (URI), z.B. Seiten, Standorte, Menschen, Produkte haben kann...
  - **Beschreibung:** Attribute, Funktionen und Beziehungen der Ressourcen
  - **Framework (Gerüst):** Modell, Sprachen und Syntaxen für diese Beschreibungen
- RDF wurde 1999 als eine **W3C Empfehlung** veröffentlicht.
  - RDF wurde ursprünglich als ein Datenmodell für **Metadaten** eingeführt.
  - RDF wurde verallgemeinert, um **Kenntnisse aller Arten** abzudecken.



## Beispiel: RDF Beschreibung einer Organisation

Nike, Dahliastraat 24, 2160 Wommelgem



```
<rdf:RDF
  xmlns:rov="http://www.w3.org/TR/vocab-regorg/"
  xmlns:org="http://www.w3.org/TR/vocab-org/"
  xmlns:locn="http://www.w3.org/ns/locn#" >

  <rov:RegisteredOrganization rdf:about="http://example.com/org/2172798119">
    <rov:legalName> "Nike" </rov:legalName>
    <org:hasRegisteredSite rdf:resource="http://example.com/site/1234"/>
  </rov:RegisteredOrganization>

  <locn:Address rdf:about="http://example.com/site/1234"/>
    <locn:fullAddress> "Dahliastraat 24, 2160 Wommelgem" </locn:fullAddress>
  </locn:Address>

</rdf:RDF>
```

# RDF-Struktur

*Triples, Grafiken und Syntax.*

## Was ist ein Triple?

*RDF ist eine allgemeine Syntax zur Darstellung von Daten im Web.*

Jede in RDF ausgedrückte Information wird als **Triple** vertreten:

- **Subjekt** – eine Quelle, die mit einer URI identifiziert werden kann.
- **Prädikat** – eine URI-identifizierte wiederverwendete Spezifikation der Beziehung.
- **Objekt** – eine Quelle oder Literale, mit dem das Thema verwandt ist.

*Beispiel: Name einer gesetzlichen Rechtspersönlichkeit:*

`http://example.com/org/2172798119 has as legal name "Nike".`

Thema

Prädikat

Objekt



# RDF wird auf Graph gegründet

**Graph =**

Eine Sammlung von Triples



# RDF Syntax

## RDF/XML

```
<rdf:RDF
```

```
  xmlns:rov="http://www.w3.org/TR/vocab-regorg/"  
  xmlns:org="http://www.w3.org/TR/vocab-org/"  
  xmlns:locn="http://www.w3.org/ns/locn#" >
```

Definition von Präfixen

```
<rov:RegisteredOrganization rdf:about="http://example.com/org/2172798119">  
  <rov:legalName> "Niké" </rov:legalName>  
  <org:hasRegisteredSite rdf:resource="http://example.com/site/1234"/>  
</rov:RegisteredOrganization>
```

```
<locn:Address rdf:about="http://example.com/site/1234"/>  
  <locn:fullAddress> "Dahliastraat 24, 2160 Wommelgem" </locn:fullAddress>  
</locn:Address>
```

Beschreibung der Daten – Triples

```
</rdf:RDF>
```

Graf



**Subjekt**

**Prädikat**

**Objekt**

RDF/XML ist derzeit die einzige Syntax, die standardisiert durch W3C ist.

# RDF Syntax

## Turtle

@prefix rov: <http://www.w3.org/TR/vocab-regorg/> .      Definition von Präfixen  
@prefix org: <http://www.w3.org/TR/vocab-org/> .  
@prefix locn: <http://www.w3.org/ns/locn#> .

< http://example.com/org/2172798119 >  
ⓐ <rov:RegisteredOrganization> ⓑ;  
  rov:legalName “Niké”;  
  org:hasRegisteredSite <http://example.com/site/1234> ⓒ

<http://example.com/site/1234>      Beschreibung der Daten – Triples  
  a <locn:Address> ;  
  locn:fullAddress “Dahliastraat 24, 2160 Wommelgem” .

Graf



**Subjekt**

**Prädikat**

**Objekt**

*Turtle wird in RDF 1.1.  
standardisiert*

**Siehe also:**

<http://www.w3.org/2009/12/rdf-ws/papers/ws11>

# RDF Syntax

## RDFa

### Einbetten von RDF Daten in HTML

```
<html>
<head> ... </head>
<body>
...
<div resource="http://example.com/org/2172798119"
typeof="http://www.w3.org/TR/vocab-regorg/RegisteredOrganization">
<p>
<span property="http://www.w3.org/TR/vocab-regorg/legalName">Nike<span>
Address: <span property="http://www.w3.org/ns/locn#fullAddress"> Dahlistraat
24, 2160 Wommelgem </span>
</p></div>
</body>
```



#### Siehe auch:

<http://www.w3.org/TR/2012/NOTE-rdfa-primer-20120607/>

# Wie repräsentieren Sie Daten in RDF?

*Klassen, Eigenschaften und Vokabular*

## *RDF Vokabular*

*“Ein Vokabular ist ein Datenmodell, das Klassen, Eigentum und Beziehungen einschließt, die zur Beschreibung Ihrer Daten und Metadaten verwendet werden können.”*

RDF Vokabulare verwenden **Sätze von Begriffen**, um Dinge zu beschreiben.

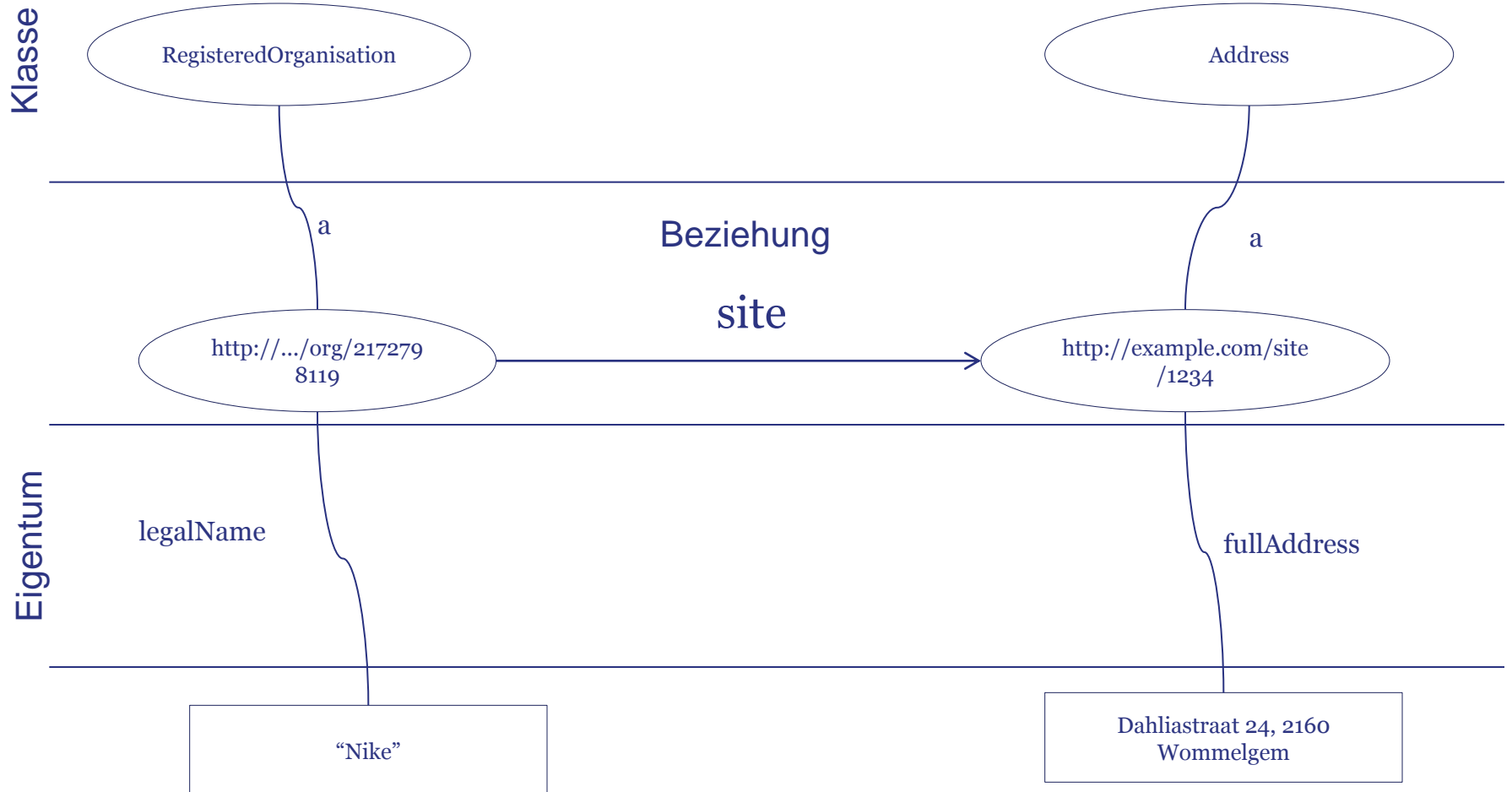
Ein Begriff ist entweder **eine Klasse oder eine Eigenschaft**.

- Objekttyp Eigenschaften (Beziehungen)
- Datentyp Eigenschaften (Attribute)

## *Was sind Klassen, Beziehungen und Eigentum?*

- **Klasse.** Ein Konstrukt, das Dinge in der realen und/oder Informationswelt repräsentiert, z.B eine Person, eine Organisation, eine Konzept wie “Gesundheit” oder “Freiheit”.
- **Beziehung.** Eine Verbindung zwischen zwei Klassen: zwischen einem Dokument und der Organisation, die es veröffentlicht (d.h. Organisation veröffentlicht Dokument); oder zwischen einer Karte und der geografischen Region, die sie zeigt (d.h. Karte zeigt geografische Region). In RDF Beziehungen werden als Objekttyp Eigenschaften kodiert.
- **Eigentum.** Eine Eigenschaft einer Klasse in einer besonderen Dimension, wie der offizielle Name einer Organisation oder das Datum und die Zeit, wann eine Beobachtung gemacht wurde.

# Beispiele von Klassen, Beziehungen und Eigentum





## Weiterverwenden von RDF Vokabularen

- Weiterverwendung ermöglicht höchste **Interoperabilität** Ihrer Daten.

*Zum Beispiel macht die Verwendung von <dcterms:created>, dessen Wert eine Datei vom Typ 2013-02-21<sup>^</sup>xsd:date sein sollte, dass die Daten sofort von viele Maschinen verarbeitbar sind. Wenn Ihr Schema Datenherausgeber dazu ermuntert, unterschiedliche Begriffe und Datumsformate, wie ex:date "21 February 2013“, zu benutzen, dann wird das Veröffentlichen von Daten mit Ihrem Schema eine weitere Verarbeitung erfordern, um das gleiche wie alle anderen zu machen.*

- Weiterverwendung **erhöht die Glaubwürdigkeit** Ihres Schemas.

*Es zeigt, dass es mit Sorge und Professionalität veröffentlicht worden ist, und das fördert wiederum seine Weiterverwendung.*

- Weiterverwendung ist **einfacher und billiger**.

*Durch die Weiterverwendung von Klassen und Eigentum mit Hilfe von gut definierte und richtig gehosteten Vokabeln wird vermieden, dass Sie diese Bemühung wiederholen müssen.*

**Siehe auch:**

<https://joinup.ec.europa.eu/community/semic/document/cookbook-translating-data-models-rdf-schemas>

<http://www.slideshare.net/OpenDataSupport/model-your-data-metadata>

# Wo können Sie vorhandene Vokabulare finden?

Linked Open Vocabularies (LOV)

The "LOV Search" Features gives you the possibility to search for an existing element (property, class or vocabulary) in the Linked Open Vocabularies Catalogue. LOV Aggregator endpoint and metrics about the use of vocabularies in the Semantic Web are used to bring you some relevant results.

Filter by Domain: 543 results in 89 vocabularies

- City (29)
- General and Meta (100)
- Library (72)
- Market (63)
- Media (13)
- Science (66)
- Web (32)
- Where & When (17)

Filter by Type

- rdfs:Class (183)
- rdf:Property (223)
- voaf:Vocabulary (54)
- Other (173)

Filter by Vocabulary (89)

URI	Label	Score
http://purl.org/rss/1.0/description (owl:DatatypeProperty)	rdfs:label Description	score:0.882
http://purl.org/dc/elements/1.1/title Description @en	rdfs:comment A short text description of the subject.	score:0.784
http://usefulinc.com/ns/doap#description (rdf:Property)	rdfs:label description @fr	score:0.756
http://purl.org/dc/terms/description (owl:ObjectProperty)	rdfs:comment Plain text description of a project, of 2-... @en	score:0.703
http://purl.org/dc/terms/description (owl:ObjectProperty)	rdfs:comment ...mment: Examples of Description include, but is not...	score:0.686
http://www.w3.org/2002/12/cal/ical#description (owl:DatatypeProperty)	rdfs:label DESCRIPTION	score:0.686
http://www.w3.org/2002/12/cal/ical#description (owl:DatatypeProperty)	rdfs:comment ...des a more complete description of the calendar com...	score:0.686
http://ivoweb.org/ontology/core#description (owl:DatatypeProperty)	rdfs:label description @en-us	score:0.686
http://rdfs/sioc/ns#description (owl:DatatypeProperty)	rdfs:label description @en	score:0.686
http://purl.org/vocab/aisso/schema#description (rdf:Property)		score:0.686

<http://lov.okfn.org/>

More targeted.

Refine the search results via the faceted search filters.

<http://joinup.ec.europa.eu/>

Joinup

Share and reuse interoperability solutions for public administrations

Advanced search

Filtering by:

- Semantic Asset Projects
- Semantic Asset Releases

Refine your search

Filter by type

- Repository of Origin
- OSLO - Open Standards for Local Administrations in Flanders (1)
- ListPoint - the open platform for code list standards (1)
- Linked Open Vocabularies (2)
- Standardisér.dk (2)
- Standards and Reports (3)
- ES1 Doc (3)
- XRepository (7)
- EU Semantic Interoperability Catalogue (12)
- ESD Standards (12)
- Dutch Standardisation Forum - "Comply or explain"-standards (50)

Search for "organisation"

1 to 20 of 112 results

Sort by: Relevancy, Popularity, Title, Author

1

2

3

More focused.

More relevant.

**Siehe auch:**

<http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/CommonVocabularies>

# Wohlbekannte Vokabulare

<u><a href="#">Friend-of-a-Friend (FOAF)</a></u>	Vokabular, um Leute zu beschreiben.
<u><a href="#">Core Person Vocabulary</a></u>	Vokabular, um die grundlegenden Eigenschaften einer Person, z.B Name, Geschlecht, Geburtsdatum zu beschreiben.
<u><a href="#">DOAP</a></u>	Vokabular, um Projekte zu beschreiben.
<u><a href="#">DCAT-AP</a></u>	Auf der Daten Katalog Vokabular (DCAT) basierte Vokabular, um Datensätzen des öffentlichen Sektors in Europa zu beschreiben.
<u><a href="#">ADMS</a></u>	Vokabular, um Interoperability Assets zu beschreiben.
<u><a href="#">Dublin Core</a></u>	Definiert allgemeine Metadatenattribute.
<u><a href="#">Registered Organisation Vocabulary</a></u>	Vokabular, um Organisationen zu beschreiben, die typisch in einem nationalen oder regionalen Register sind.
<u><a href="#">Organization Ontology</a></u>	Vokabular zur Beschreibung der Struktur von Organisationen.
<u><a href="#">Core Location Vocabulary</a></u>	Vokabular, das die grundlegenden Eigenschaften eines Standortes beschreibt.
<u><a href="#">Core Public Service Vocabulary</a></u>	Vokabular, das die grundlegenden Eigenschaften eines Dienstes der öffentlichen Verwaltung beschreibt.
<u><a href="#">schema.org</a></u>	Vokabular, das Von Google, Yahoo und Microsoft verstanden wird, um strukturierte Daten im Web zu veröffentlichen.

# Modellieren Sie Ihre eigenes Vokabular als ein RDF Schema

Wenn es kein geeignetes maßgebliches wiederverwendbares Vokabular gibt, um Ihre Daten zu beschreiben, verwenden Sie Konventionen, um Ihr eigenes Vokabular zu schreiben.

- RDF Schema (RDFS)
- Web Ontology Language (OWL)

## Beispiel: Definition einer Klasse

```
cpsv:PublicService a rdfs:Class, owl:Class;  
  rdfs:label "Public Service"@en;  
  rdfs:comment "This class represents the service itself. As noted in  
the scope, a public service is the capacity to carry out a procedure  
and exists whether it is used or not. It is a set of deeds and  
acts performed by or on behalf of a public agency for the benefit of a  
citizen, a business or another public agency."@en.
```

### Siehe auch:

<http://www.slideshare.net/OpenDataSupport/model-your-data-metadata>

# Einführung in SPARQL

*Die RDF Query-Sprache*

# Über SPARQL

*SPARQL ist die Standardsprache, um als RDF Triple verfasste Graph-Daten abzufragen.*

## SPARQL Protocol And RDF Query Language

- Einer der drei Kernstandards des Semantic Web, zusammen mit RDF und OWL.
- Wurde im Januar 2008 ein W3C-Standard.
- SPARQL 1.1 befindet sich derzeit im Arbeitsversion-Status.

## Arten von SPARQL Abfragen

- **SELECT**  
Gibt eine Tabelle aller X, Y, usw. an, die die folgenden Bedingungen erfüllen...
- **CONSTRUCT**  
Findet alle X, Y, usw., die die folgenden Bedingungen erfüllen... Und setzt sie in die folgende Schablone ein, um (möglicherweise neue) RDF Anweisungen zu generieren und eine neue Grafik zu schaffen.
- **DESCRIBE**  
Findet alle Anweisungen im Datensatz, die Information über die folgende(n) Quelle(n) liefern... (identifiziert durch Name oder Beschreibung).
- **ASK**  
Gibt es irgendwelche X,Y, usw., die die folgenden Bedingungen erfüllen...?

**Siehe auch:**

<http://www.euclid-project.eu/modules/chapter2>

# Struktur einer SPARQL Abfrage

Typ der  
Anfrage

PREFIX rov: <http://www.w3.org/TR/vocab-regorg/>

Definition von  
Präfixen

SELECT ?name Variablen, d.h. was zu suchen ist.

WHERE

{ ?x rov:legalName ?name }

RDF Triple Muster, d.h.  
die Bedingungen, die  
erfüllt werden müssen.



# *SELECT – Rückschluss auf den Namen einer Organisation mit bestimmten URI*

## Beispieldaten

```
comp:A rov:haslegalName "Nike" .
comp:A org:hasRegisteredSite site:1234 .

Comp:B rov:haslegalName "BARCO" .

site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem .
```

## Abfrage

```
PREFIX comp: < http://example.org/org/>
PREFIX org: < http://www.w3.org/TR/vocab-regorg/ >
PREFIX site: <http://example.org/site/>
PREFIX rov: <http://www.w3.org/TR/vocab-regorg/>

SELECT ?name

WHERE
{ ?x org:hasRegisteredSite site:1234 .
  ?x rov:haslegalName ?name .}
```

## Ergebnis

**name**

"Nike"

# SELECT - Rückschluss auf Namen und Adresse von Organisationen

## Beispieldaten

```
comp:A rov:haslegalName "Niké" .  
comp:A org:hasRegisteredSite site:1234 .
```

```
Comp:B rov:haslegalName "BARCO" .
```

```
site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem" .
```

## Abfrage

```
PREFIX org: < http://www.w3.org/TR/vocab-regorg/ >  
PREFIX locn: < http://www.w3.org/ns/locn# >  
PREFIX rov: < http://www.w3.org/TR/vocab-regorg/ >
```

```
SELECT ?name ?address
```

```
WHERE
```

```
{ ?x org:hasRegisteredSite ?site.  
  ?x rov:haslegalName ?name .  
  ?site locn:fullAddress ?address . }
```

## Ergebnis

name	address
"Niké"	"Dahliastraat 24, 2160 Wommelgem"

# *SELECT – Rückschluss auf alle Bücher mit einem bestimmten Preis (1/2)*

## Beispieldaten

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

```
@prefix : <http://example.org/book/> .
```

```
@prefix ns: <http://example.org/ns#> .
```

```
:book1 dc:title "SPARQL Tutorial" .
```

```
:book1 ns:price 42 .
```

```
:book2 dc:title "The Semantic Web" .
```

```
:book2 ns:price 23 .
```

## *SELECT – Rückschluss auf alle Bücher unterhalb eines bestimmten Preises (2/2)*

### Abfrage

```
PREFIX dc: <http://purl.org/dc/elements/1.1/> .  
PREFIX : <http://example.org/book/> .  
PREFIX ns: <http://example.org/ns#> .
```

```
SELECT ?book ?title
```

### WHERE

```
{ ?book dc:title ?title .  
  ?book ns:price ?price . FILTER ( ?price < 40 )  
}
```

### Ergebnis

<b>book</b>	<b>title</b>
:book2	“The Semantic Web”

# CONSTRUCT – Erstellt eine neue Grafik für den Vergleich mit einer anderen Marke

## Beispieldaten

```
comp:A rov:haslegalName "Niké" .  
comp:A org:hasRegisteredSite site:1234 .  
  
comp:B rov:haslegalName "BARCO" .  
  
site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem" .
```

## Abfrage

```
PREFIX comp: < http://example/org/org/>  
PREFIX org: < http://www.w3.org/TR/vocab-regorg/ >  
PREFIX rdfs: <http://www.w3.org/TR/rdf-schema/>  
  
CONSTRUCT {?comp rdfs:label ?name}  
  
WHERE  
{ ?comp org:haslegalName ?name. }
```

## resultierender Graph

```
@prefix comp: <http://example/org/> .  
@prefix rdfs: <http://www.w3.org/TR/rdf-schema/>  
  
comp:a rdfs:label "Niké" .  
comp:b rdfs:label "BARCO" .
```

# DESCRIBE – Rückschluss auf alle Triples von an einem bestimmten Standort registrierten Organisationen

## Beispieldaten

```
comp:A rov:haslegalName "Niké" .
comp:A org:hasRegisteredSite site:1234 .

comp:B rov:haslegalName "BARCO" .

site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem" .
```

## Abfrage

```
PREFIX comp: <http://example/org/>
PREFIX site: <http://example/site>
PREFIX org: < http://www.w3.org/TR/vocab-regorg/

DESCRIBE ?organisation

WHERE
{?organisation org:hasRegisteredSite site:1234}
```

## Ergebnis

```
@prefix comp: <http://example/org/> .
@prefix org: <http://www.w3.org/TR/vocab-regorg/> .

comp:A has:legalName "Niké" .
comp:A org:hasRegisteredSite site:1234 .
```

## *DESCRIBE – Rückschluss auf alle Triples, die mit einer bestimmten Quelle (Organisation) verbunden sind.*

### Beispieldaten

```
comp:A rov:haslegalName "Niké" .  
comp:A org:hasRegisteredSite site:1234 .  
  
comp:B rov:haslegalName "BARCO" .  
  
site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem" .
```

### Abfrage

```
PREFIX comp: <http://example/org/>  
  
DESCRIBE comp:A
```

### Ergebnis

```
@prefix comp: <http://example/org/> .  
@prefix org: <http://www.w3.org/TR/vocab-regorg/> .  
  
comp:A rov:haslegalName "Niké" .  
comp:A org:hasRegisteredSite site:1234 .
```

# ASK – Gibt es Organisationen, deren eingetragener Standort “1234” ist?

## Beispieldaten

```
comp:A rov:haslegalName “Niké” .  
comp:A org:hasRegisteredSite site:1234 .
```

```
comp:B rov:haslegalName “BARCO” .
```

```
site:1234 locn:fullAddress “Dahliastraat 24, 2160 Wommelgem” .
```

## Abfrage

```
PREFIX org: < http://www.w3.org/TR/vocab-regorg/
```

```
ASK
```

```
WHERE
```

```
{?organisation org:hasRegisteredSite site:1234}
```

## Ergebnis

```
TRUE
```



# ASK – Gibt es einen registrierten Standort für die Organisation “BARCO”?

## Beispieldaten

```
comp:A rov:haslegalName “Niké” .  
comp:A org:hasRegisteredSite site:1234 .
```

```
comp:B rov:haslegalName “BARCO” .
```

```
site:1234 locn:fullAddress “Dahliastraat 24, 2160 Wommelgem” .
```

## Abfrage

```
PREFIX comp: <http://example/org/>  
PREFIX org: <http://www.w3.org/TR/vocab-regorg/>
```

```
ASK
```

```
WHERE
```

```
{comp:B org:hasRegisteredSite ?site .}
```

## Ergebnis

```
FALSE
```

# SPARQL 1.1 Update

Einsetzbar für...

- Hinzufügen von Daten (INSERT)
- Löschen von Daten (DELETE)
- Laden einer RDF Grafik (LOAD / LOAD .. INTO)
- Löschen einer RDF Grafik (CLEAR GRAPH)
- Erstellen einer RDF Grafik (CREATE GRAPH)
- Entfernen einer RDF Grafik (DROP GRAPH)
- Kopieren einer RDF Grafik (COPY GRAPH ... TO GRAPH)
- Verschieben einer RDF Grafik (MOVE GRAPH ... TO GRAPH)
- Hinzufügen einer RDF Grafik (ADD GRAPH TO GRAPH)

**Siehe auch:**

<http://www.euclid-project.eu/modules/chapter2>

<http://www.w3.org/TR/sparql11-update/>

## *INSERT – existiert ein registrierter Standort für “BARCO”?*

### Beispieldaten

```
comp:A rov:haslegalName “Niké” .  
comp:A org:hasRegisteredSite site:1234 .  
  
comp:B rov:haslegalName “BARCO” .  
  
site:1234 locn:fullAddress “Dahliastraat 24, 2160 Wommelgem” .
```

### Abfrage

```
PREFIX comp: <http://example/org/>  
PREFIX org: <http://www.w3.org/TR/vocab-regorg/>  
  
INSERT DATA  
{  
  site:5678 locn:fullAddress “President Kennedypark 35, 8500 Kortrijk” .  
  comp:B org:hasRegisteredSite site:5678 .  
}
```

### Ergebnis

```
comp:A rov:haslegalName “Niké” .  
comp:A org:hasRegisteredSite site:1234 .  
  
comp:B rov:haslegalName “BARCO” .  
comp:B org:hasRegisteredSite site:5678 .  
  
site:1234 locn:fullAddress “Dahliastraat 24, 2160 Wommelgem” .  
site:5678 locn:fullAddress “President Kennedypark 35, 8500 Kortrijk” .
```

# INSERT/DELETE – Ändert die Adresse für “Niké”?

## Data

```
comp:A rov:haslegalName “Niké” .
comp:A org:hasRegisteredSite site:1234 .

comp:B rov:haslegalName “BARCO” .

site:1234 locn:fullAddress “Dahliastraat 24, 2160 Wommelgem” .
```

## Abfrage

```
PREFIX comp: <http://example/org/>
PREFIX org: <http://www.w3.org/TR/vocab-regorg/>
```

### DELETE DATA

```
{
comp:A org:hasRegisteredSite site:1234 .
}
```

### INSERT DATA

```
{
site:5678 locn:fullAddress “Rue de Loi 34, 1000 Bruxelles” .
comp:A org:hasRegisteredSite site:5678 .
}
```

## Ergebnis

```
comp:A rov:haslegalName “Niké” .
comp:A org:hasRegisteredSite site:1000.

site:1234 locn:fullAddress “Dahliastraat 24, 2160 Wommelgem” .
site:1000 locn:fullAddress “Rue de Loi 34, 1000 Bruxelles” .
```

# Schlussfolgerungen

- RDF ist eine allgemeine Weise, um **Daten**, die für die Veröffentlichung im **Web** vorgesehen sind, auszudrücken.
- RDF Daten werden in **Triples** ausgedrückt: Subjekt, Prädikat, Objekt.
- **Verschiedene Syntaxen** existieren, um Daten in RDF auszudrücken.
- SPARQL ist eine standardisierte Sprache, um Grafik Daten, die als RDF ausgedrückt werden, abzufragen.
- SPARQL kann verwendet werden, um RDF Daten abzufragen und zu aktualisieren.

# Vielen Dank!

## ... Und jetzt IHRE Fragen?

### Nehmen Sie auch den Online-Test!

# *Diese Präsentation wurde von Open Data Support erstellt*

## **Disclaimers**

1. The views expressed in this presentation are purely those of the authors and may not, in any circumstances, be interpreted as stating an official position of the European Commission. The European Commission does not guarantee the accuracy of the information included in this presentation, nor does it accept any responsibility for any use thereof.

Reference herein to any specific products, specifications, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favouring by the European Commission.

All care has been taken by the author to ensure that s/he has obtained, where necessary, permission to use any parts of manuscripts including illustrations, maps, and graphs, on which intellectual property rights already exist from the titular holder(s) of such rights or from her/his or their legal representative.

2. This presentation has been carefully compiled by PwC, but no representation is made or warranty given (either express or implied) as to the completeness or accuracy of the information it contains. PwC is not liable for the information in this presentation or any decision or consequence based on the use of it. PwC will not be liable for any damages arising from the use of the information contained in this presentation. The information contained in this presentation is of a general nature and is solely for guidance on matters of general interest. This presentation is not a substitute for professional advice on any particular matter. No reader should act on the basis of any matter contained in this publication without considering appropriate professional advice.

Autoren:

Michiel De Keyzer, Nikolaos Loutas, Christophe Colas and Stijn Goedertier

# Referenzen

Folie 6:

- Semantic Web Stack. W3C. <http://www.w3.org/DesignIssues/diagrams/sweb-stack/2006a.png>

Folien 18 & 20:

- Linked Data Cookbook. W3C. [http://www.w3.org/2011/gld/wiki/Linked\\_Data\\_Cookbook](http://www.w3.org/2011/gld/wiki/Linked_Data_Cookbook)

Folie 21:

- Cookbook for translating data models to RDF schemas. ISA Programme. <https://joinup.ec.europa.eu/community/semic/document/cookbook-translating-data-models-rdf-schemas>

Folie 22:

- Common Vocabularies / Ontologies / Micromodels. W3C. <http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/CommonVocabularies>

Folien 23-24:

- SPARQL Query Language for RDF. W3C. <http://www.w3.org/TR/rdf-sparql-query/>

Folie 24:

- Module 2: Querying Linked Data. EUCLID. <http://www.euclid-project.eu/modules/course2>

Folie 35:

- Module 2: Querying Linked Data. EUCLID. <http://www.euclid-project.eu/modules/course2>
- SPARQL 1.1 Update. W3C.. <http://www.w3.org/TR/sparql11-update/>

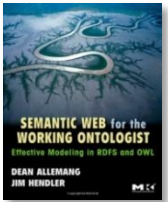


## Weiterführende Lektüre



Learning SPARQL. Bob DuCharme.

<http://www.learningsparql.com/>



Semantic Web for the working ontologist. Dean Allemang, Jim Hendler.

<http://workingontologist.org/>



EUCLID - Course 2: Querying Linked Data

<http://www.euclid-project.eu/modules/course2>

## Verwandte Projekte und Initiativen

joinup

Joinup, <https://joinup.ec.europa.eu/>



Linked Open Vocabularies, <http://okfn.org/>



W3C GLD WG, [http://www.w3.org/2011/gld/wiki/Main\\_Page](http://www.w3.org/2011/gld/wiki/Main_Page)  
W3C Schools – Learn RDF  
<http://www.w3schools.com/rdf/default.asp>



EUCLID, <http://euclid-project.eu/>



TopBraid Composer



Protégé Ontology Editor , <http://protege.stanford.edu/>



XML Summer School <http://xmlsummerschool.com/>



# Werden Sie Teil unseres Teams...

*Finden Sie  
uns auf*



Open Data Support

<http://www.slideshare.net/OpenDataSupport>



Open Data Support

<http://goo.gl/y9ZZI>

*Folgen Sie  
uns auf*



@OpenDataSupport

*Begleiten Sie  
uns auf*



joinup

<http://www.opendatasupport.eu>

*Kontaktieren  
Sie uns unter*

[contact@opendatasupport.eu](mailto:contact@opendatasupport.eu)