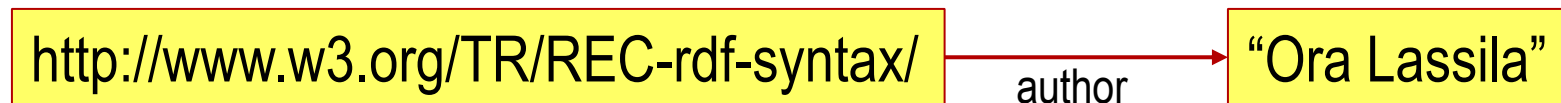# RDF & SPARQL

Instructor: Peter Baumann

email: p.baumann@jacobs-university.de
tel: -3178
office: room 88, Research 1

RDF: Resource Description Framework

- Framework for resources on the Web = very simple assertion language

  - Conceptual model: directed, labeled graphs
  - designed for computers, not people
  - part of W3C's Semantic Web Activity
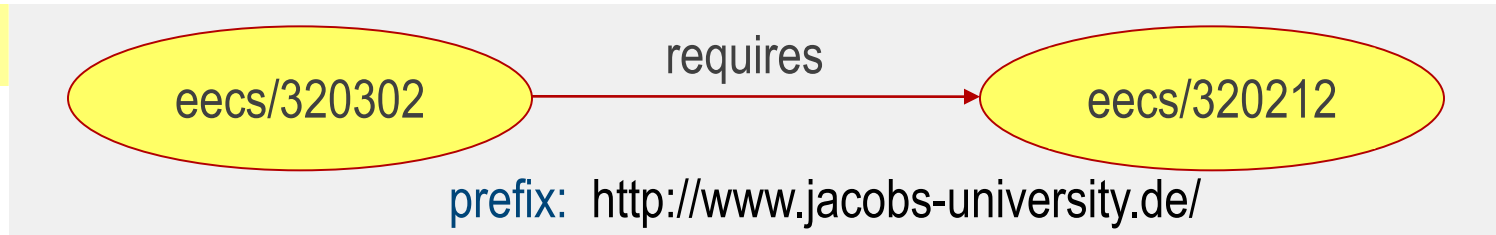
- RDF statement = resource with properties with values

```
 resource  ──property──▶  value
```

- Ex: "*http://www.w3.org/TR/REC-rdf-syntax/ has the author Ora Lassila*"

```
 http://www.w3.org/TR/REC-rdf-syntax/  ──author──▶  "Ora Lassila"
```

Alternative RDF Representations

Graphs:

eecs/320302 —requires→ eecs/320212

prefix: http://www.jacobs-university.de/

Notation3:

```
@prefix in:http://www.iu-bremen.de/ .
in:eecs/320302 in:requires in:eecs/320212 .
```

compact & readable alternative to RDF's XML syntax

XML:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:in="http://www.iu-bremen.de/">
    <rdf:description rdf:about="http://www.jacobs-university.de/eecs/320302">
        <in:requires rdf:resource="http://www.jacobs-university.de/eecs/320212"/>
    </rdf:description>
</rdf:RDF>
```

RDF doc root

{}:

{ "http://www. jacobs-university.de/eecs/320302", requires,
  "http://www. jacobs-university.de/eecs/320212" }

SPARQL

- = <u>S</u>imple <u>P</u>rotocol and <u>R</u>DF <u>Q</u>uery <u>L</u>anguage

  - W3C recommendation

- = QL for RDF

  - extract information in the form of URIs, blank nodes, plain and typed literals

  - extract RDF subgraphs

  - construct new RDF graphs based on information in the queried graphs

- *Let's taste the flavor...*

From SQL To SPARQL

SELECT name, email

FROM Person

From SQL To SPARQL

JACOBS UNIVERSITY

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

From SQL To SPARQL
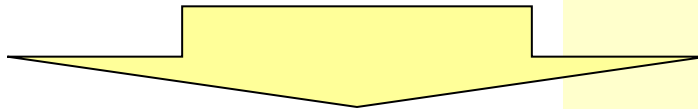
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name ?email

WHERE {

   ?person   rdf:type   foaf:Person.

   ?person   foaf:name   ?name.

   ?person   foaf:mbox   ?email.

}

SPARQL Example

- Data
  (1 triple):

  <http://example.org/book/book1>
  <http://purl.org/dc/elements/1.1/title>
  "SPARQL Tutorial" .

- Query:

  SELECT ?title
  WHERE
      {
          <http://example.org/book/book1>
          <http://purl.org/dc/elements/1.1/title>
          ?title .

      }

  title
  "SPARQL Tutorial" .