



UNIVERSIDADE  
FEDERAL DO CEARÁ

# Como Publicar Dados Relacionais em RDF

Luís Eufrazio T. Neto

Mestrado em Ciência da Computação

Departamento de Computação

Orientadora: Dra. Vânia Vidal

# Introdução

- Linked Data: padrão estabelecido como a melhor prática para expor, compartilhar e conectar dados na Web Semântica.
- Grande maioria dos dados corporativos estão armazenados em SGBDs relacionais.
- É preciso publicar os dados relacionais usando o modelo RDF.
- Como fazer isso?

# Introdução

- A solução mais utilizada é criar mapeamentos de bases relacionais para ontologias.
- Ferramentas que utilizam essa abordagem: D2R Server, OpenLink Virtuoso e Spyder.
- Existe uma linguagem recomendada pela w3c para criação dos mapeamentos: R2RML.
- As ferramentas deverão suportar R2RML.

# Conteúdo

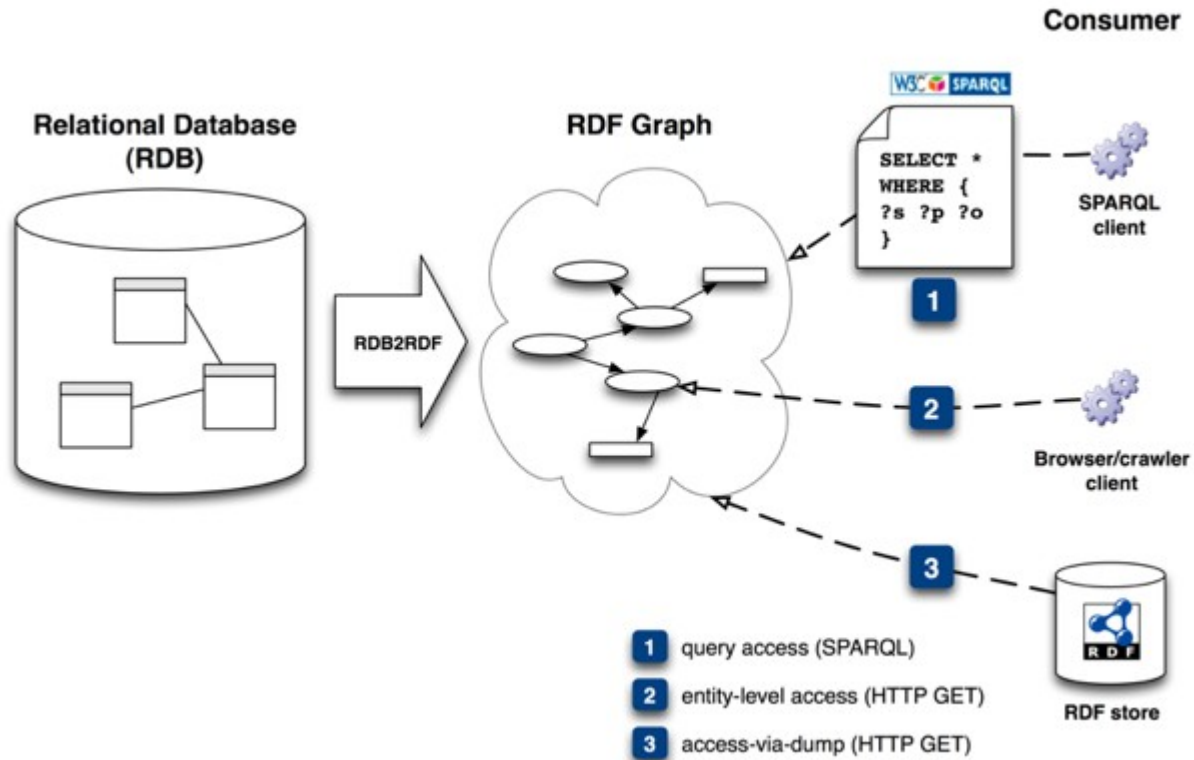
- RDB2RDF
  - Mapeamento Direto (Direct Mapping)
  - Mapeamento com reuso de vocabulários
- Plataforma D2RQ
  - Arquitetura
  - Utilização (principais comandos)
  - Linguagem de mapeamento
- A Linguagem R2RML: recomendação da w3c

**RDB2RDF**

# RDB2RDF

- Bancos relacionais são populares porque são eficientes e possuem definições precisas.
- RDF é um formato de dados que possui uma arquitetura escalável na Web para identificação e interpretação de termos.
- Desafio: mapear esquemas relacionais estruturados e com restrições em um modelo de dados simples, baseado em Grafo, flexível e com o mínimo de restrições.

# RDB2RDF



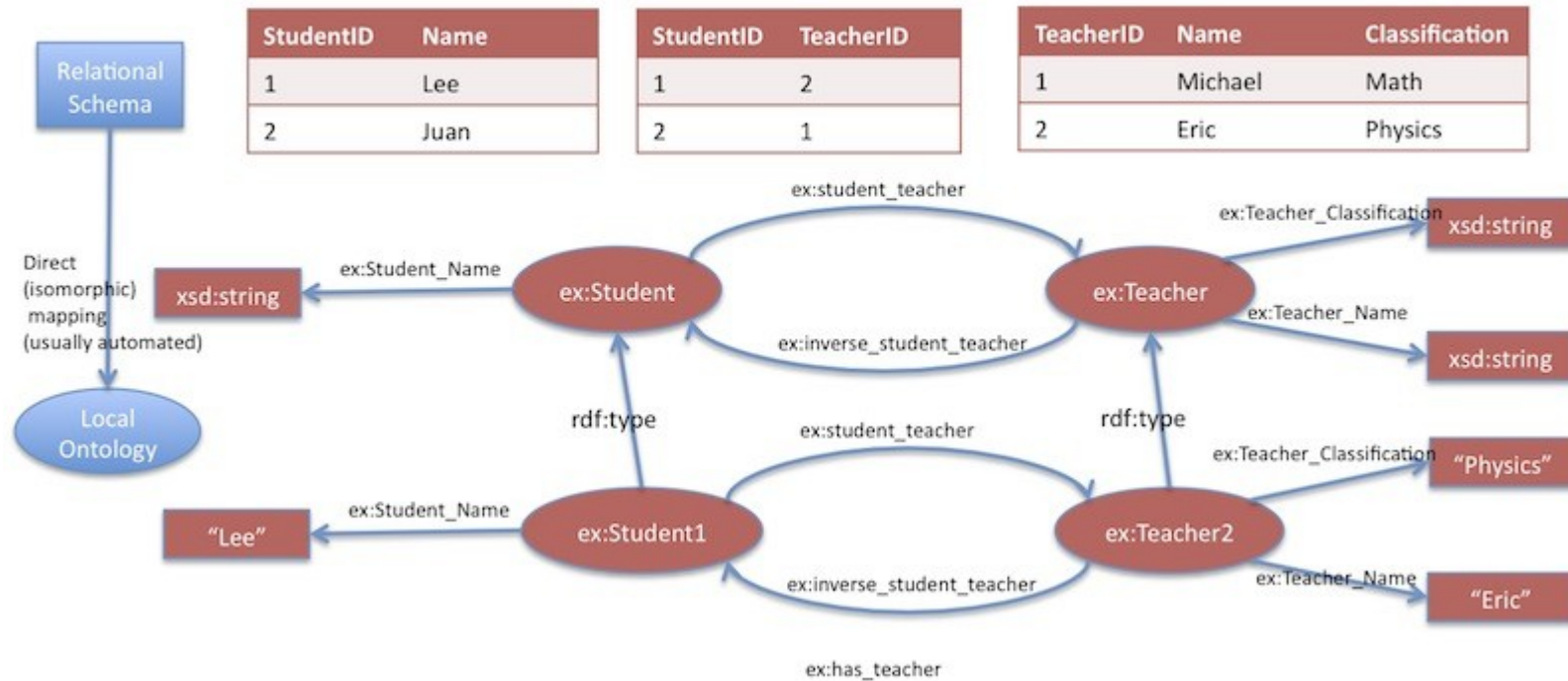
# Direct Mapping - DM

- Define uma transformação simples que serve como base para definição de transformações mais complexas.
- Provê o comportamento default para os mapeamentos criados usando R2RML.
- Recebe como entrada um banco de dados relacional (dados e esquema), e gera um grafo RDF que é chamado de **direct graph**.



# Direct Mapping - DM

Option 1: Direct Mapping (no domain ontology involved)



# Direct Mapping - DM

- Exemplo simples para introduzir os conceitos de mapeamento direto
- SQL (DDL) para criar um exemplo simples com duas tabelas que têm uma coluna como chave primária e uma chave estrangeira fazendo o relacionamento entre elas:

# Direct Mapping - DM

```
CREATE TABLE "Addresses" (  
  "ID" INT, PRIMARY KEY("ID"),  
  "city" CHAR(10),  
  "state" CHAR(2)  
)  
  
CREATE TABLE "People" (  
  "ID" INT, PRIMARY KEY("ID"),  
  "fname" CHAR(10),  
  "addr" INT,  
  FOREIGN KEY("addr") REFERENCES "Addresses"("ID")  
)  
  
INSERT INTO "Addresses" ("ID", "city", "state") VALUES (18, 'Cambridge', 'MA')  
INSERT INTO "People" ("ID", "fname", "addr") VALUES (7, 'Bob', 18)  
INSERT INTO "People" ("ID", "fname", "addr") VALUES (8, 'Sue', NULL)
```

People			Addresses		
PK		→ Address(ID)	PK		
ID	fname	addr	ID	city	state
7	Bob	<a href="#">18</a>	18	Cambridge	MA
8	Sue	NULL			

```
@base <http://foo.example/DB/> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
  
<People/ID-7> rdf:type <People> .  
<People/ID-7> <People#ID> 7 .  
<People/ID-7> <People#fname> "Bob" .  
<People/ID-7> <People#addr> 18 .  
<People/ID-7> <People#ref-addr> <Addresses/ID-18> .  
<People/ID-8> rdf:type <People> .  
<People/ID-8> <People#ID> 8 .  
<People/ID-8> <People#fname> "Sue" .  
  
<Addresses/ID-18> rdf:type <Addresses> .  
<Addresses/ID-18> <Addresses#ID> 18 .  
<Addresses/ID-18> <Addresses#city> "Cambridge" .  
<Addresses/ID-18> <Addresses#state> "MA" .
```

# Direct Mapping - DM

- Esquemas mais complexos incluem chaves compostas.

```
CREATE TABLE "Addresses" (  
    "ID" INT,  
    "city" CHAR(10),  
    "state" CHAR(2),  
    PRIMARY KEY("ID")  
)  
  
CREATE TABLE "Department" (  
    "ID" INT,  
    "name" CHAR(10),  
    "city" CHAR(10),  
    "manager" INT,  
    PRIMARY KEY("ID"),  
    UNIQUE ("name", "city")  
)  
  
CREATE TABLE "People" (  
    "ID" INT,  
    "fname" CHAR(10),  
    "addr" INT,  
    "deptName" CHAR(10),  
    "deptCity" CHAR(10),  
    PRIMARY KEY("ID"),  
    FOREIGN KEY("addr") REFERENCES "Addresses"("ID"),  
    FOREIGN KEY("deptName", "deptCity") REFERENCES "Department"("name", "city")  
)  
  
ALTER TABLE "Department" ADD FOREIGN KEY("manager") REFERENCES "People"("ID")
```

# Direct Mapping - DM

People

PK		→ <i>Addresses(ID)</i>	→ <i>Department(name, city)</i>	
ID	fname	addr	deptName	deptCity
7	Bob	<u>18</u>	<u>accounting</u>	<u>Cambridge</u>
8	Sue	NULL	NULL	NULL

Addresses

PK		
ID	city	state
18	Cambridge	MA

Department

PK	Unique Key		→ <i>People(ID)</i>
ID	name	city	manager
23	accounting	Cambridge	<u>8</u>

```

@base <http://foo.example/DB/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<People/ID-7> rdf:type <People> .
<People/ID-7> <People#ID> 7 .
<People/ID-7> <People#fname> "Bob" .
<People/ID-7> <People#addr> 18 .
<People/ID-7> <People#ref-addr> <Addresses/ID-18> .
<People/ID-7> <People#deptName> "accounting" .
<People/ID-7> <People#deptCity> "Cambridge" .
<People/ID-7> <People#ref-deptName.deptCity> <Department/ID-23> .
<People/ID-8> rdf:type <People> .
<People/ID-8> <People#ID> 8 .
<People/ID-8> <People#fname> "Sue" .

<Addresses/ID-18> rdf:type <Addresses> .
<Addresses/ID-18> <Addresses#ID> 18 .
<Addresses/ID-18> <Addresses#city> "Cambridge" .
<Addresses/ID-18> <Addresses#state> "MA" .

<Department/ID-23> rdf:type <Department> .
<Department/ID-23> <Department#ID> 23 .
<Department/ID-23> <Department#name> "accounting" .
<Department/ID-23> <Department#city> "Cambridge" .
<Department/ID-23> <Department#manager> 8 .
<Department/ID-23> <Department#ref-manager> <People#ID-8> .
    
```

# Reuso de Vocabulários

- Reuse, não reinvente
  - foaf:name ao invés de ex:fname
  - rdf:type no lugar de ex:type
- Como encontrar termos já existentes
  - Sindice.com (web search)
  - UMBEL Subject Finder (web services)
- Crie Links
  - Use os predicados owl:sameAs e rdfs:seeAlso

# Reuso de Vocabulários

```
@base <http://foo.example/DB/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<People/ID-7> rdf:type foaf:Person .
<People/ID-7> <People#ID> 7 .
<People/ID-7> foaf:name "Bob" .
<People/ID-7> <People#addr> 18 .
<People/ID-7> <People#ref-addr> <Addresses/ID-18> .
<People/ID-8> rdf:type <People> .
<People/ID-8> <People#ID> 8 .
<People/ID-8> <People#fname> "Sue" .

<Addresses/ID-18> rdf:type <Addresses> .
<Addresses/ID-18> <Addresses#ID> 18 .
<Addresses/ID-18> <Addresses#city> "Cambridge" .
<Addresses/ID-18> <Addresses#state> "MA" .
```

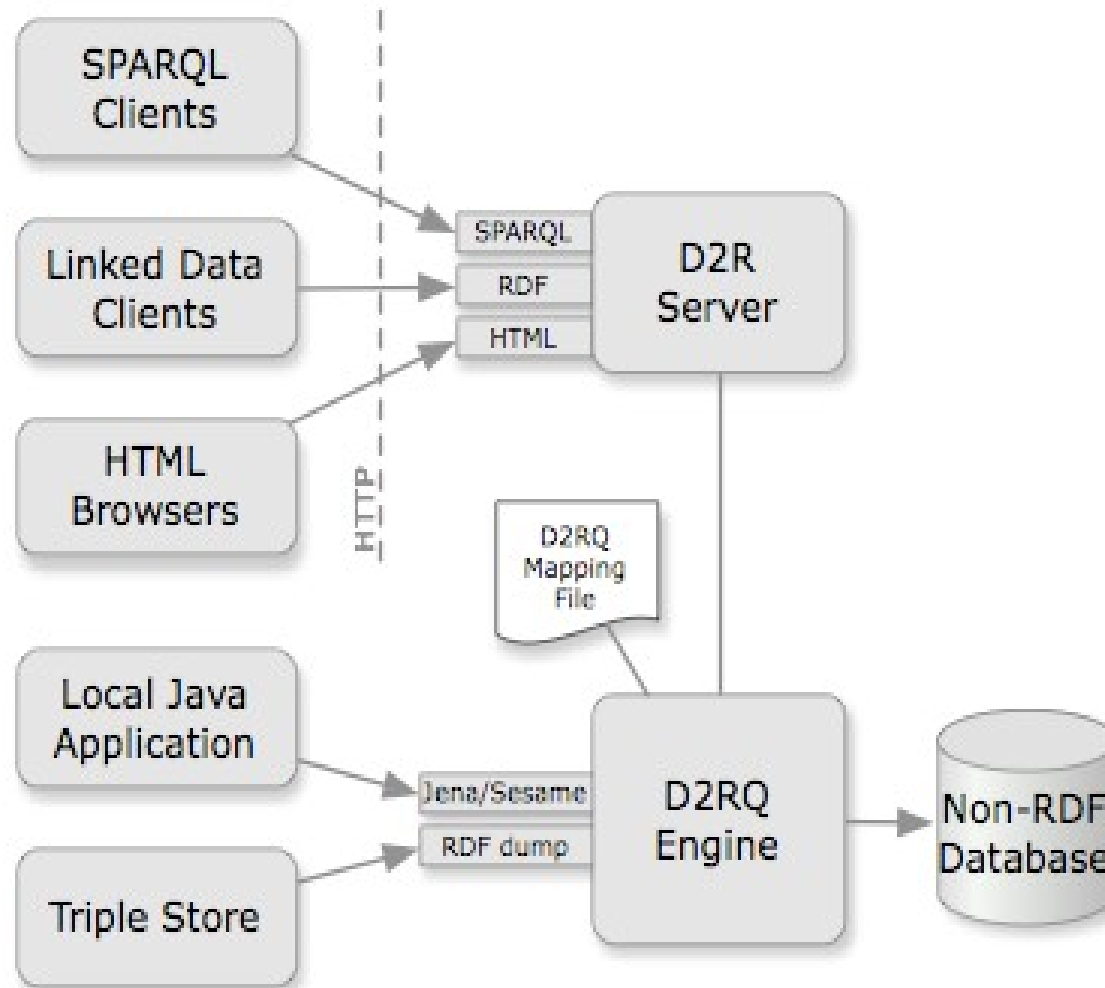
# **A Plataforma D2RQ**



# Plataforma D2RQ

- É um sistema para acessar dados relacionais na forma de datasets RDF virtuais (grafos RDF que não podem ser alterados).
- A plataforma consiste de:
  - D2RQ Mapping Language
  - D2RQ Engine
  - D2R Server

# Plataforma D2RQ



# Plataforma D2RQ

- SGBDs suportados
  - Oracle
  - MySQL
  - PostgreSQL
  - SQL Server
  - HSQLDB
  - Interbase/Firebird
  - Access?

# **Principais Comandos do D2RQ**

# D2RQ Principais Comandos

- generate-mapping
- dump-rdf
- d2r-server

# D2RQ Generate Mapping

- Script para criar um arquivo de mapeamento D2RQ
- Mapeamento direto ou default mapping
- O mapeamento gerado pode ser utilizado como está ou pode ser customizado

# D2RQ Generate Mapping

`generate-mapping`

`[-u user]`

`[-p password]`

`[-d driver]`

`[-o outfile.ttl]`

`[-b baseURI]`

`[-s schema]`

`[-v]`

`jdbcURL`

# D2RQ Generate Mapping

## Banco Local:

```
generate-mapping -u root jdbc:mysql:///iswc
```

## Banco Remoto:

```
generate-mapping
```

```
-u riccyg -p password
```

```
-d oracle.jdbc.OracleDriver
```

```
-o staffdb-mapping.ttl
```

```
jdbc:oracle:thin:@ora.intranet.derri.ie:1521:staffdb
```



# D2RQ Dump RDF

- Script que triplifica todos os dados do banco relacional em um único arquivo RDF
- Pode ser feito com ou sem arquivo de mapeamento
- Se nenhum arquivo de mapeamento for passado, o script generate-mapping é invocado internamente, gerando um mapeamento direto

# D2RQ Dump RDF

Com arquivo de mapeamento:

```
dump-rdf
```

```
-m mapping.ttl [output parameters]
```

```
[output parameters]
```

```
-f format
```

```
-b baseURI
```

```
-o outfile
```

# D2RQ Dump RDF

Sem arquivo de mapeamento:

```
dump-rdf -j jdbcURL
```

```
[-d driverclass] [-u username]
```

```
[-p password] [output parameters]
```

```
[output parameters]
```

```
-f format
```

```
-b baseURI
```

```
-o outfile
```

# D2RQ Dump RDF

- Com arquivo de mapeamento:

```
dump-rdf -m mapping-iswc.ttl -f N-  
TRIPLE -b http://localhost:2020/
```

- Sem arquivo de mapeamento:

```
dump-rdf -u root  
-j jdbc:mysql:///iswc  
-f RDF/XML-ABBREV -o iswc-dump.rdf
```

# D2R Server

- Aplicação web para que publica bancos de dados relacionais na Web Semântica
- Browsers html podem navegar no grafo RDF gerado
- Permite consultas usando a linguagem SPARQL.

# D2R Server

- Executa de 3 formas:
  - Linha de comando
    - `d2r-server [-p port] [-b serverBaseURI] [--fast] mapping-file.ttl`
    - recomenda-se para testes
  - Deploy em um container web
    - Aplicação web
    - Em produção
  - Como um serviço do Windows
    - `install-service servicename mapping.ttl`

# D2R Server

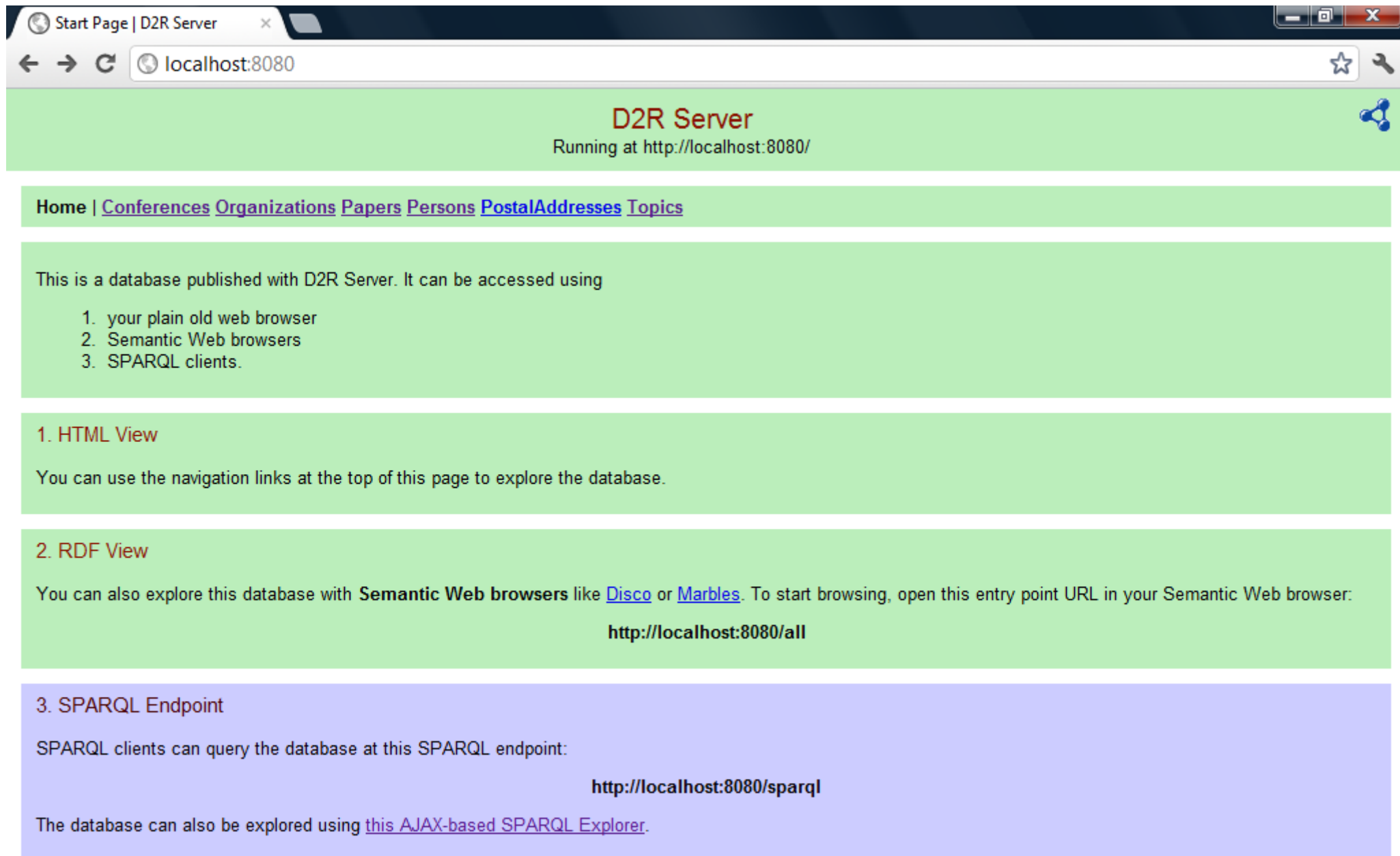
- Exemplo:

```
d2r-server
```

```
-p 8080
```

```
iswc.ttl
```

# D2R Server



Start Page | D2R Server

localhost:8080

## D2R Server

Running at <http://localhost:8080/>

[Home](#) | [Conferences](#) [Organizations](#) [Papers](#) [Persons](#) [PostalAddresses](#) [Topics](#)

This is a database published with D2R Server. It can be accessed using

1. your plain old web browser
2. Semantic Web browsers
3. SPARQL clients.

### 1. HTML View

You can use the navigation links at the top of this page to explore the database.

### 2. RDF View

You can also explore this database with **Semantic Web browsers** like [Disco](#) or [Marbles](#). To start browsing, open this entry point URL in your Semantic Web browser:

**<http://localhost:8080/all>**

### 3. SPARQL Endpoint

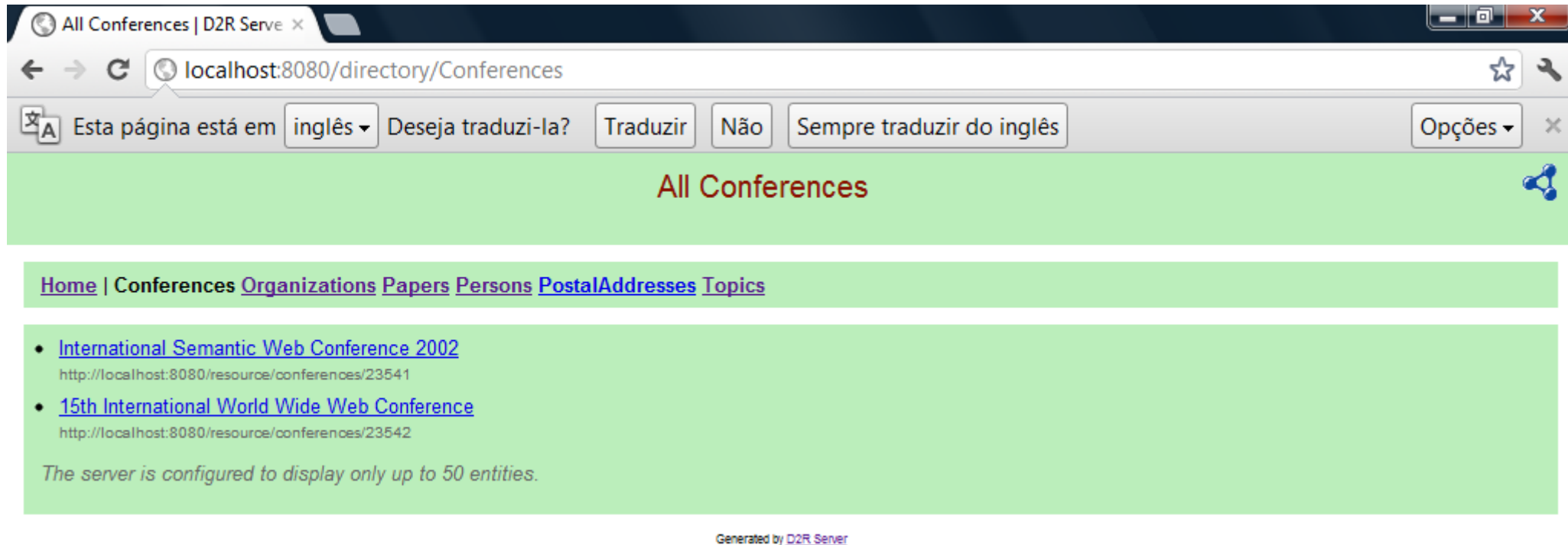
SPARQL clients can query the database at this SPARQL endpoint:

**<http://localhost:8080/sparql>**

The database can also be explored using [this AJAX-based SPARQL Explorer](#).



# D2R Server



All Conferences | D2R Serve x

localhost:8080/directory/Conferences

Esta página está em inglês Deseja traduzi-la? Traduzir Não Sempre traduzir do inglês Opções x

All Conferences

[Home](#) | [Conferences](#) [Organizations](#) [Papers](#) [Persons](#) [PostalAddresses](#) [Topics](#)

- [International Semantic Web Conference 2002](http://localhost:8080/resource/conferences/23541)  
http://localhost:8080/resource/conferences/23541
- [15th International World Wide Web Conference](http://localhost:8080/resource/conferences/23542)  
http://localhost:8080/resource/conferences/23542

*The server is configured to display only up to 50 entities.*

Generated by [D2R Server](#)

# D2R Server

International Semantic Web x

localhost:8080/page/conferences/23541

Esta página está em inglês ▾ Deseja traduzi-la? Traduzir Não Sempre traduzir do inglês Opções ▾

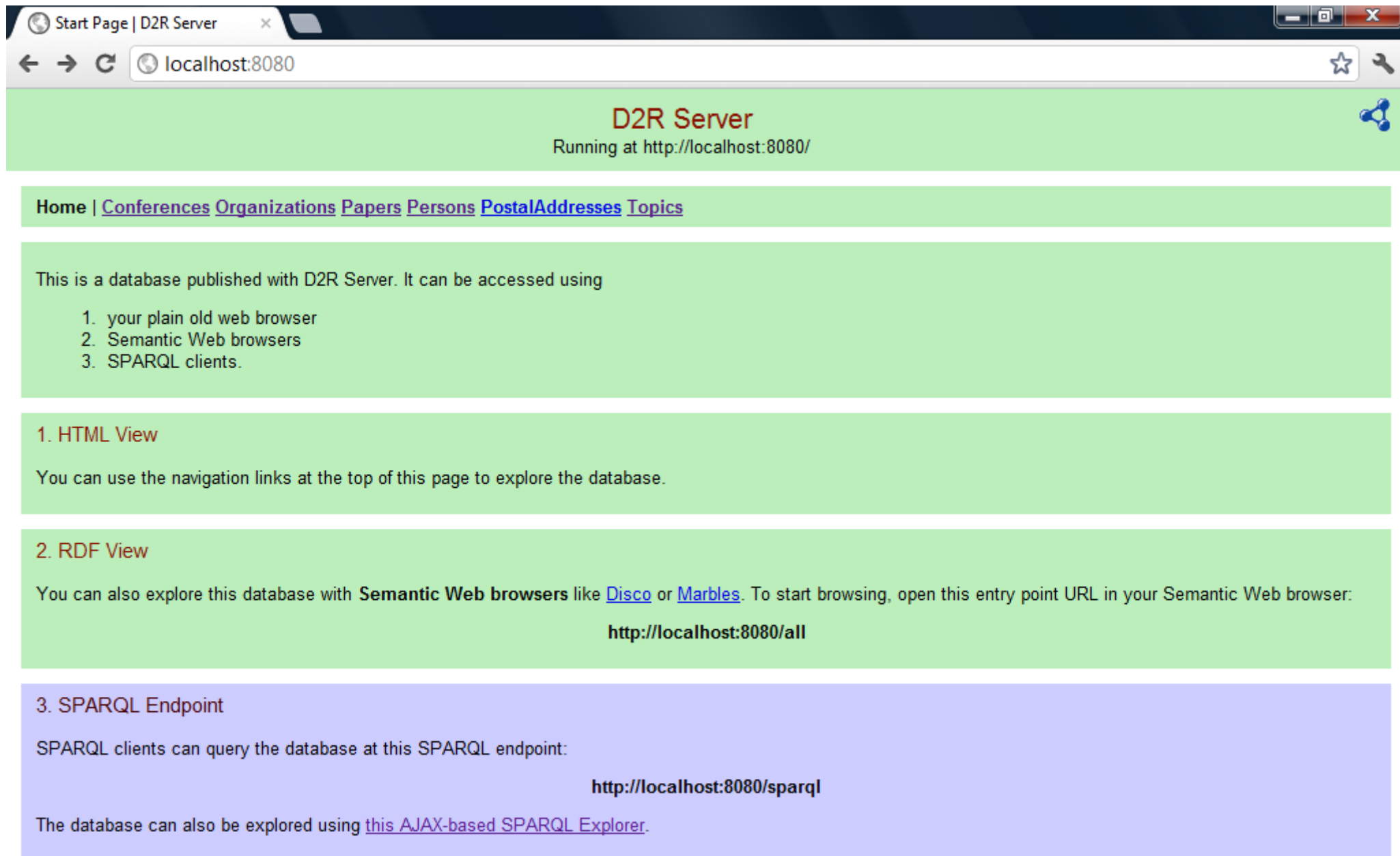
**International Semantic Web Conference 2002**  
Resource URI: <http://localhost:8080/resource/conferences/23541>

[Home](#) | [All Conferences](#)

Property	Value
is iswc:conference of	< <a href="http://localhost:8080/resource/papers/1">http://localhost:8080/resource/papers/1</a> >
is iswc:conference of	< <a href="http://localhost:8080/resource/papers/2">http://localhost:8080/resource/papers/2</a> >
is iswc:conference of	< <a href="http://localhost:8080/resource/papers/4">http://localhost:8080/resource/papers/4</a> >
is iswc:conference of	< <a href="http://localhost:8080/resource/papers/5">http://localhost:8080/resource/papers/5</a> >
dc:date	June 9-12, 2002
rdfs:label	International Semantic Web Conference 2002
iswc:location	Sardinia
vocab:startDate	2002-10-09T00:00:00 (xsd:dateTime)
rdf:type	iswc:Conference

*The server is configured to display only a limited number of values (limit per property bridge: 50).*

# D2R Server



Start Page | D2R Server x

localhost:8080

## D2R Server

Running at <http://localhost:8080/>

[Home](#) | [Conferences](#) [Organizations](#) [Papers](#) [Persons](#) [PostalAddresses](#) [Topics](#)

This is a database published with D2R Server. It can be accessed using

1. your plain old web browser
2. Semantic Web browsers
3. SPARQL clients.

### 1. HTML View

You can use the navigation links at the top of this page to explore the database.

### 2. RDF View

You can also explore this database with **Semantic Web browsers** like [Disco](#) or [Marbles](#). To start browsing, open this entry point URL in your Semantic Web browser:

**<http://localhost:8080/all>**

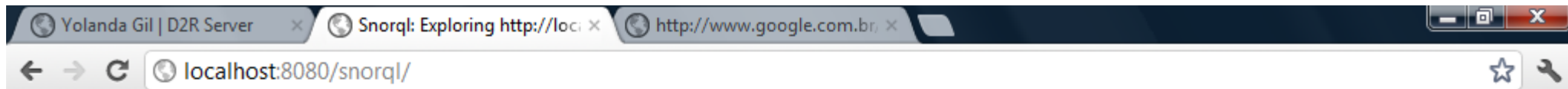
### 3. SPARQL Endpoint

SPARQL clients can query the database at this SPARQL endpoint:

**<http://localhost:8080/sparql>**

The database can also be explored using [this AJAX-based SPARQL Explorer](#).

# D2R Server



## Snorql: Exploring http://localhost:8080/sparql

### SPARQL:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX iswc: <http://annotation.semanticweb.org/iswc/iswc.daml#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX map: <file:/Users/richard/D2RQ/workspace/D2RQ/doc/example/mapping-iswc.ttl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vocab: <http://localhost:8080/resource/vocab/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
```

```
SELECT DISTINCT * WHERE {
  ?s ?p ?o
}
LIMIT 10
```

Results:

### Browse:

- [Classes](#)
- [Properties](#)

# D2R Server

Semantic Annotation | D2R x Snorql: Exploring http://loci x

localhost:8080/snorql/?query=SELECT+DISTINCT+%3Flabel+WHERE+%7B%0D%0A++%3Ftopic+skos%3AprefLabel+%3Flabel%  
Esta página está em inglês ▾ Deseja traduzi-la? Traduzir Não Sempre traduzir do inglês Opções ▾

Snorql: Exploring http://localhost:8080/sparql

## SPARQL:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX iswc: <http://annotation.semanticweb.org/iswc/iswc.daml#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX map: <file:/Users/richard/D2RQ/workspace/D2RQ/doc/example/mapping-iswc.ttl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vocab: <http://localhost:8080/resource/vocab/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
```

```
SELECT DISTINCT ?label WHERE {
  ?topic skos:prefLabel ?label
  FILTER regex(?label, "^Semantic")
}
LIMIT 10
```

Results: Browse ▾ Go! Reset

## Browse:

- [Classes](#)
- [Properties](#)

## SPARQL results:

label
"Semantic Annotation"^^xsd:string
"Semantic Web"^^xsd:string
"Semantic Web Languages"^^xsd:string
"Semantic Web Infrastructure"^^xsd:string

# **D2RQ Mapping Language**

# D2RQ Mapping Language

- Linguagem de mapeamentos declarativa para mapear esquemas relacionais em vocabulários RDF e ontologias OWL
- Escrita usando a sintaxe Turtle
- O mapeamento usa termos do namespace:  
<http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#>

# D2RQ Mapping Language

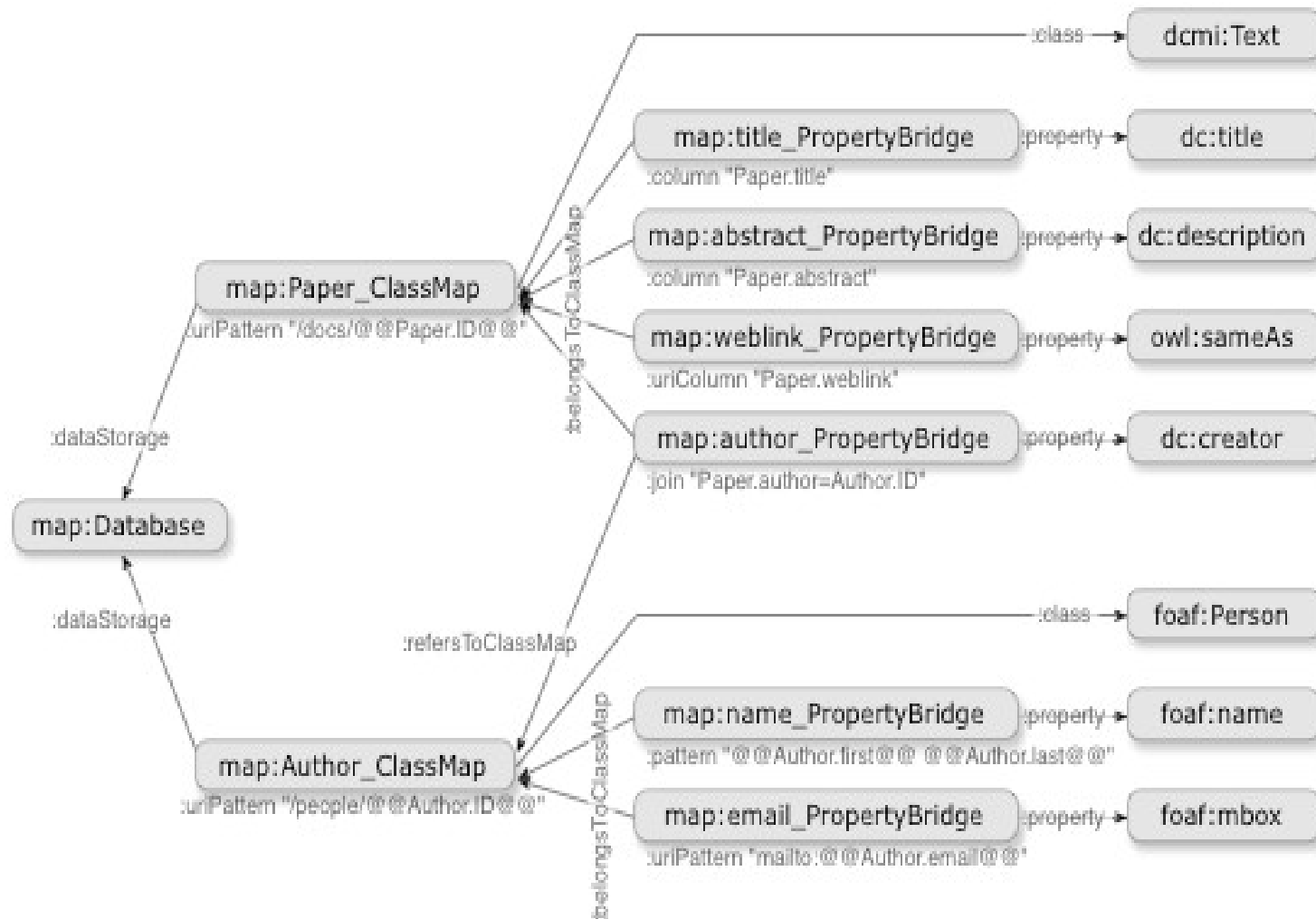
- Os termos do namespace estão formalmente definidos em duas versões Turtle e RDF/XML:
  - <http://d2rq.org/terms/d2rq.ttl>
  - <http://d2rq.org/terms/d2rq.rdf>



# D2RQ Mapping Language

- Os termos do namespace estão formalmente definidos em duas versões Turtle e RDF/XML:
  - <http://d2rq.org/terms/d2rq.ttl>
  - <http://d2rq.org/terms/d2rq.rdf>
- Um mapeamento define um grafo RDF virtual que contém informações de um banco de dados
  - conceito parecido com o de visões SQL

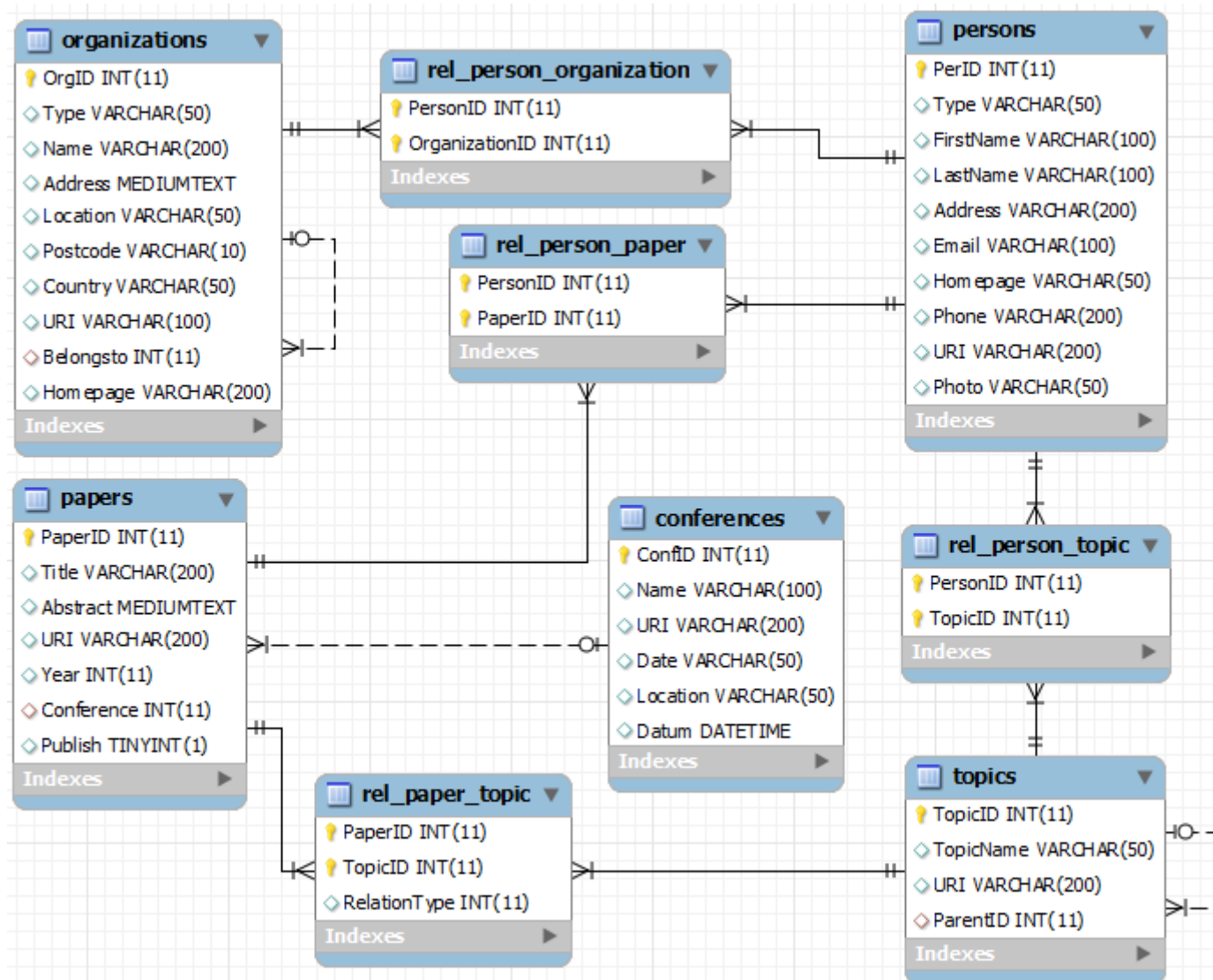
# D2RQ Mapping Language



# D2RQ Mapping Language

- Exemplo: ISWC
  - Banco de dados que armazena informações sobre conferências, papers, autores e tópicos.
  - Ontologia ISWC
    - <http://d2rq.org/example/iswc.daml>
  - Arquivo de mapeamento
    - <http://d2rq.org/example/mapping-iswc.ttl>
  - Dump do banco
    - <http://d2rq.org/example/iswc-mysql.sql>

# D2RQ Mapping Language



# D2RQ Mapping Language

```
# D2RQ Namespace
@prefix d2rq:      <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
# Namespace of the ontology
@prefix : <http://annotation.semanticweb.org/iswc/iswc.daml#> .

# Namespace of the mapping file; does not appear in mapped data
@prefix map: <file:///Users/d2r/example.ttl#> .

# Other namespaces
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

# D2RQ Mapping Language

```
map:Database1 a d2rq:Database;  
    d2rq:jdbcDSN "jdbc:mysql://localhost/iswc";  
    d2rq:jdbcDriver "com.mysql.jdbc.Driver";  
    d2rq:username "user";  
    d2rq:password "password";  
    .
```

# D2RQ Mapping Language

```
# -----  
# CREATE TABLE Conferences (ConfID int, Name text, Location text);  
  
map:Conference a d2rq:ClassMap;  
    d2rq:dataStorage map:Database1.  
    d2rq:class :Conference;  
    d2rq:uriPattern "http://conferences.org/comp/confno@@Conferences.ConfID@@";  
    .  
map:eventTitle a d2rq:PropertyBridge;  
    d2rq:belongsToClassMap map:Conference;  
    d2rq:property :eventTitle;  
    d2rq:column "Conferences.Name";  
    d2rq:datatype xsd:string;  
    .  
map:location a d2rq:PropertyBridge;  
    d2rq:belongsToClassMap map:Conference;  
    d2rq:property :location;  
    d2rq:column "Conferences.Location";  
    d2rq:datatype xsd:string;  
    .
```

# D2RQ Mapping Language

- ClassMap
  - representa uma classe de uma ontologia OWL ou esquema RDFS
  - conectado a um d2rq:Database
  - tem um conjunto de d2rq:PropertyBridges
- 4 mecanismos para atribuir identificadores



# D2RQ Mapping Language

## 1. URI Patterns

`http://example.org/persons/@@Persons.ID@@`

`http://example.org/lineItems/item@@Orders.orderID@@-@@LineItems.itemID@@`

`urn:isbn:@@Books.isbn@@`

`mailto:@@Persons.email@@aa`

- usados com a propriedade `d2rq:uriPattern`

# D2RQ Mapping Language

- Para caracteres não permitidos em URIs
  - @@Table.Column|urlencode@@
  - @@Table.Column|urlify@@
    - Url encode + espaços → \_
- URI pattern relativa
  - gera URI relativa
  - deve ser combinada com a URI base para formar URIs completas
  - Exemplo: persons/@@Persons.ID@@

# D2RQ Mapping Language

```
map:PaperClassMap a d2rq:ClassMap;  
  d2rq:uriPattern "http://www.conference.org/conf02004/paper#Paper@@Papers.PaperID@@";  
  d2rq:class :Paper;  
  d2rq:classDefinitionLabel "paper"@en;  
  d2rq:classDefinitionComment "A conference paper."@en;  
  d2rq:dataStorage map:Database1;  
  .
```

# D2RQ Mapping Language

## 2. URI columns

- colunas no banco de dados que referenciam URIs
- `d2rq:uriColumn "TableName.ColumnName"`

## 3. URI expressions

- expressão SQL para formar a URI
- `d2rq:uriSqlExpression`  
`"CONCAT('http://www.company.com/homepages/', user.username)"`

# D2RQ Mapping Language

## 4. Blank Nodes

- para recursos que existem e possuem propriedades, mas não possuem nomes para identificá-los
- d2rq:bNodeIdColumns
  - Lista de "TableName.ColumnName" separados por vírgula

# D2RQ Mapping Language

```
map:Topic a d2rq:ClassMap;  
    d2rq:bNodeIdColumns "Topics.TopicID";  
    d2rq:class :Topic;  
    d2rq:classDefinitionLabel "topic"@en;  
    d2rq:classDefinitionComment "A topic."@en;  
    d2rq:dataStorage map:Database1;  
    .
```

# D2RQ Mapping Language

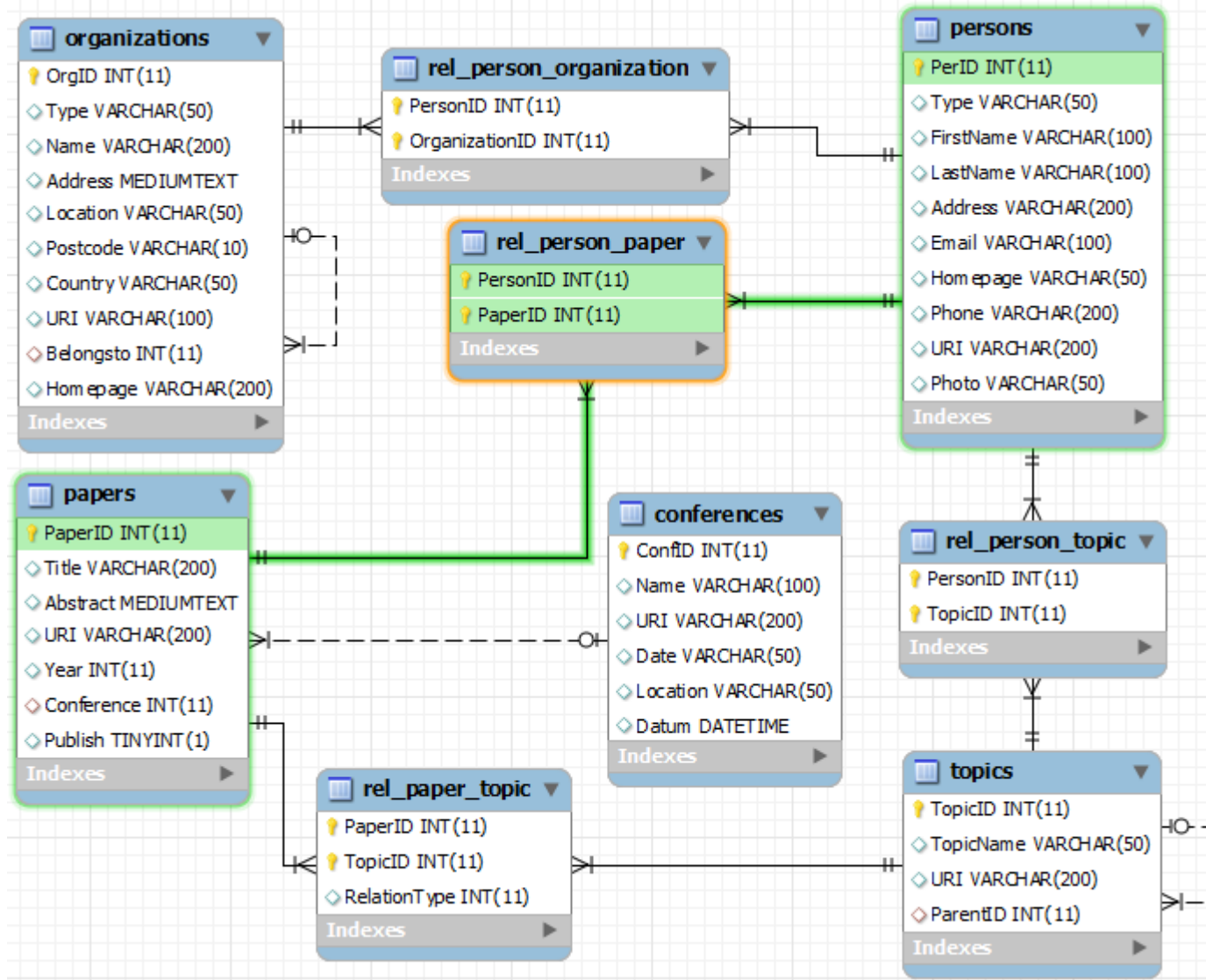
- PropertyBridge
  - cria a correspondência entre uma coluna Relacional com uma propriedade RDF.
  - pertence a um ClassMap
  - seus valores podem ser literais, mas também podem ser URIs ou blank nodes
  - valores NULL → propriedade não criada

# D2RQ Mapping Language

```
map:PaperTitle a d2rq:PropertyBridge;  
  d2rq:belongsToClassMap map:Paper;  
  d2rq:property :title;  
  d2rq:column "Papers.Title";  
  d2rq:lang "en";  
  d2rq:propertyDefinitionLabel "title"@en;  
  d2rq:propertyDefinitionComment "A paper's title."@en;  
  .
```



# D2RQ Mapping Language

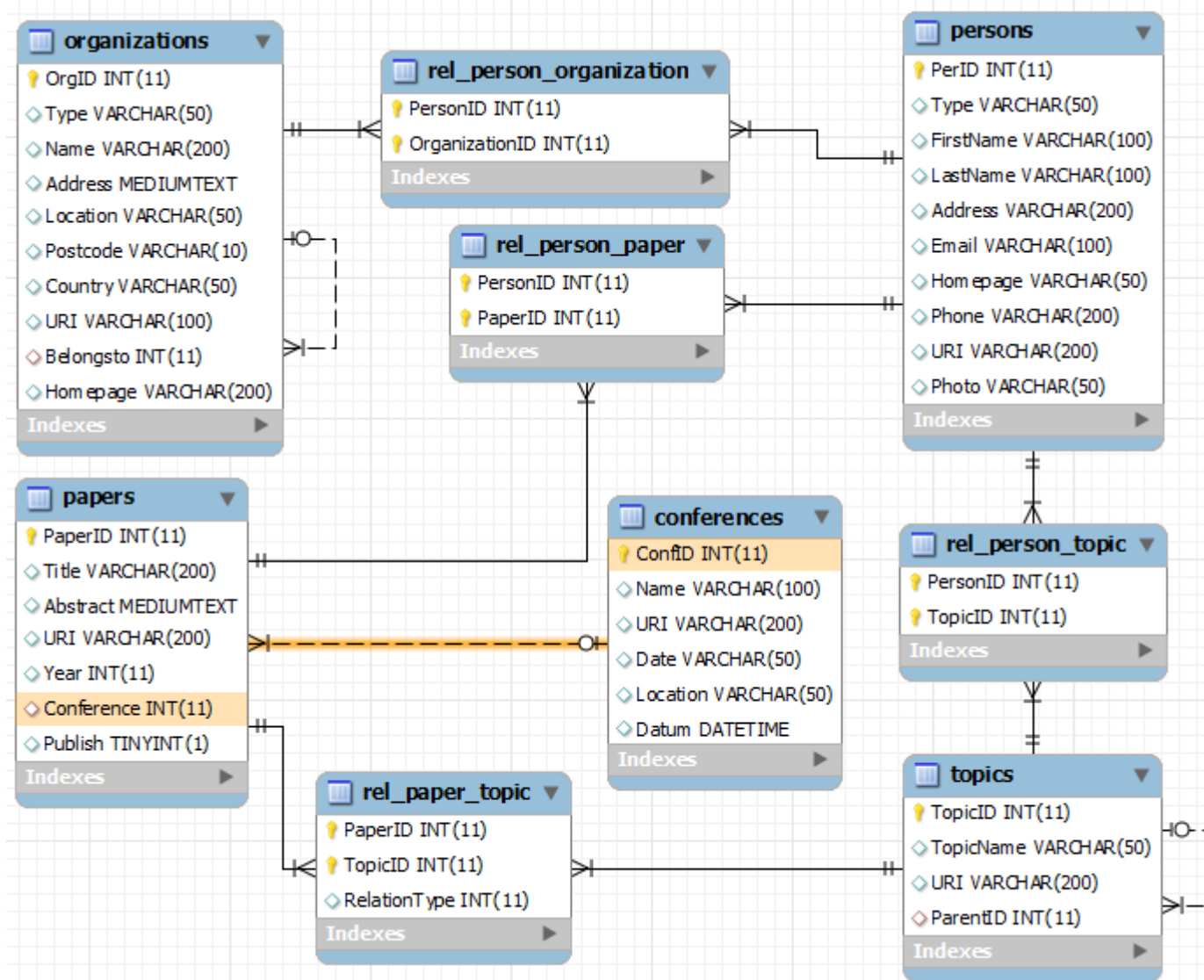


# D2RQ Mapping Language

- PropertyBridge com relacionamento N:N

```
map:authorName a d2rq:PropertyBridge;  
  d2rq:belongsToClassMap map:Papers;  
  d2rq:property :authorName;  
  d2rq:column "Persons.Name";  
  d2rq:join "Papers.PaperID <= Rel_Person_Paper.PaperID";  
  d2rq:join "Rel_Person_Paper.PersonID => Persons.PerID";  
  d2rq:datatype xsd:string;  
  d2rq:propertyDefinitionLabel "name"@en;  
  d2rq:propertyDefinitionComment "Name of an author."@en;  
  .
```

# D2RQ Mapping Language

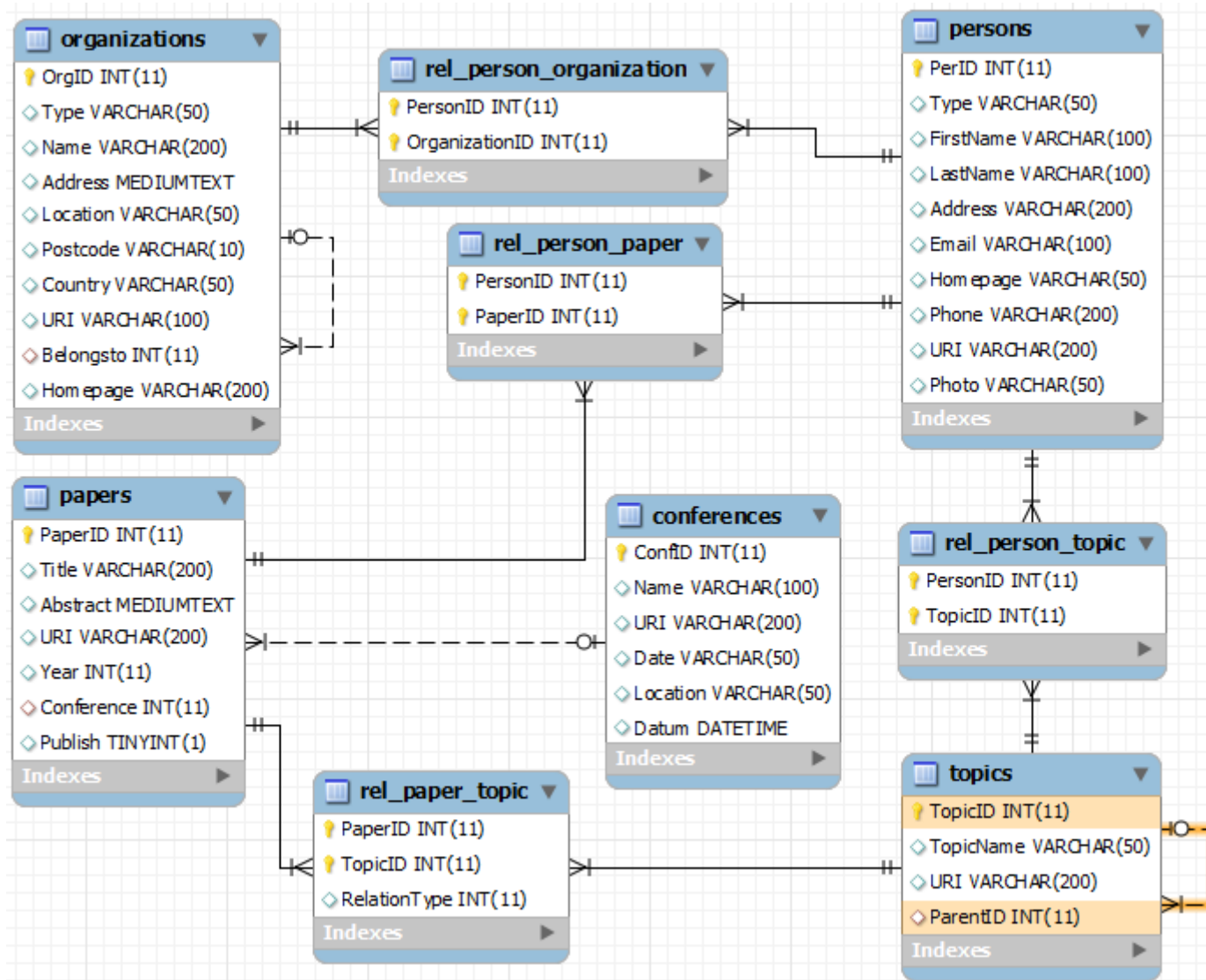


# D2RQ Mapping Language

- PropertyBridge relacionamento 1:N

```
map:PaperConference a d2rq:PropertyBridge;  
  d2rq:belongsToClassMap map:Paper;  
  d2rq:property :conference;  
  d2rq:refersToClassMap map:Conference;  
  d2rq:join "Papers.Conference => Conferences.ConfID"  
  .
```

# D2RQ Mapping Language



# D2RQ Mapping Language

- PropertyBridge relacionamento 1:N

```
map:ParentTopic a d2rq:PropertyBridge;  
  d2rq:belongsToClassMap map:Topic;  
  d2rq:property :parentTopic;  
  d2rq:refersToClassMap map:Topic;  
  d2rq:join "Topics.ParentID => ParentTopics.ID";  
  d2rq:alias "Topics AS ParentTopics";  
  .
```

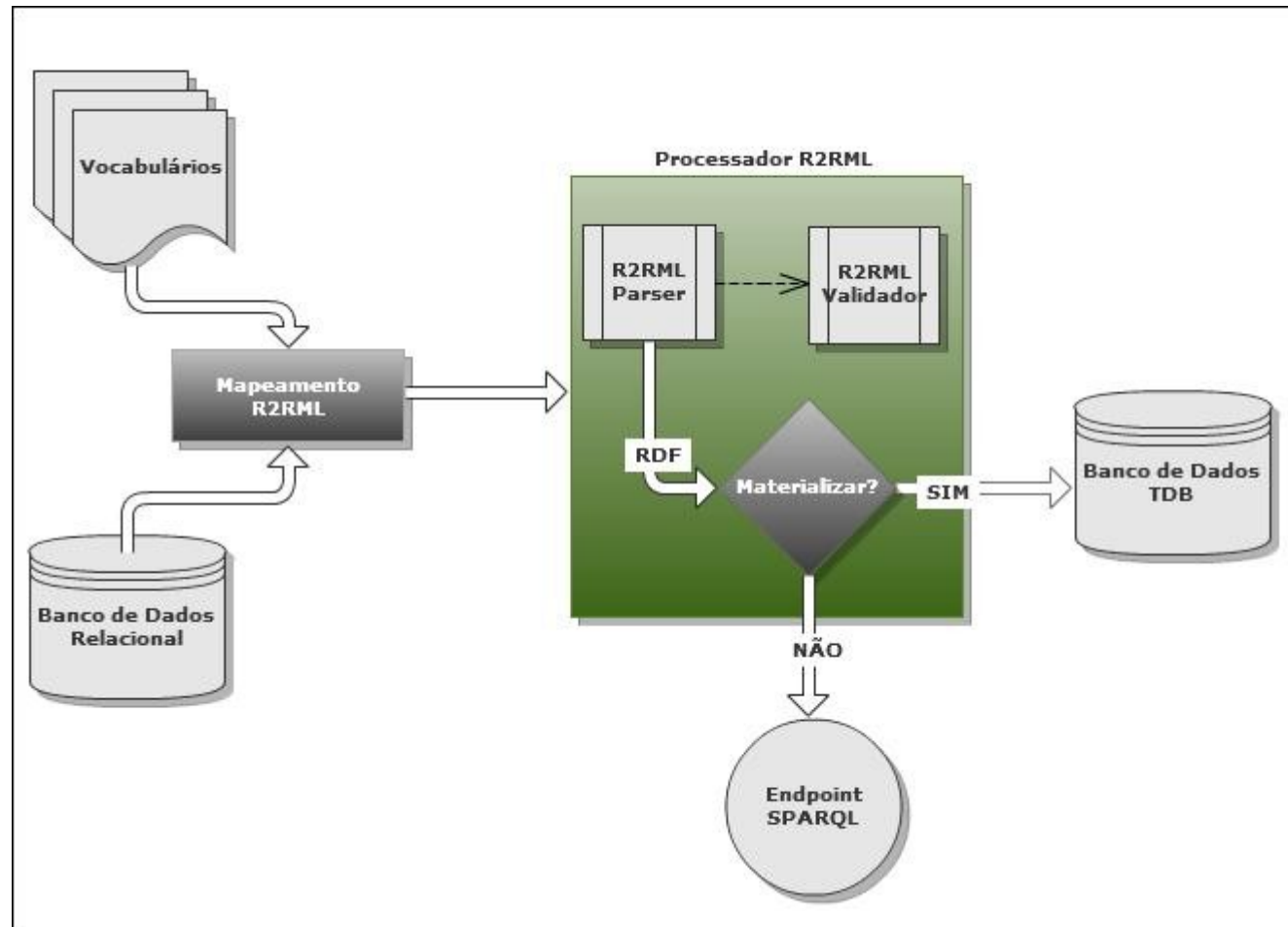
**R2RML**

# R2RML

- É uma linguagem para expressar mapeamentos customizados de bases relacionais para datasets RDF.
- Cada mapeamento é criado para um esquema de banco de dados específico e para um vocabulário alvo.



# R2RML



# R2RML

Um mapeamento faz referência a tabelas lógicas que podem ser de 3 tipos:

1. Uma tabela do esquema do banco relacional;
2. Uma visão do esquema do banco relacional;
3. Uma consulta SQL válida chamada de “Visão R2RML”, criada dentro do mapeamento.

# R2RML

- Cada tabela lógica é mapeada para RDF pela criação do elemento *TriplesMap*.
- Cada *TriplesMap* é uma regra que mapeia as tuplas da sua tabela lógica para uma ou mais triplas RDF.
- Um *TriplesMap* é composto de um *SubjectMap* e de um ou mais *PredicateObjectMap*.

# R2RML

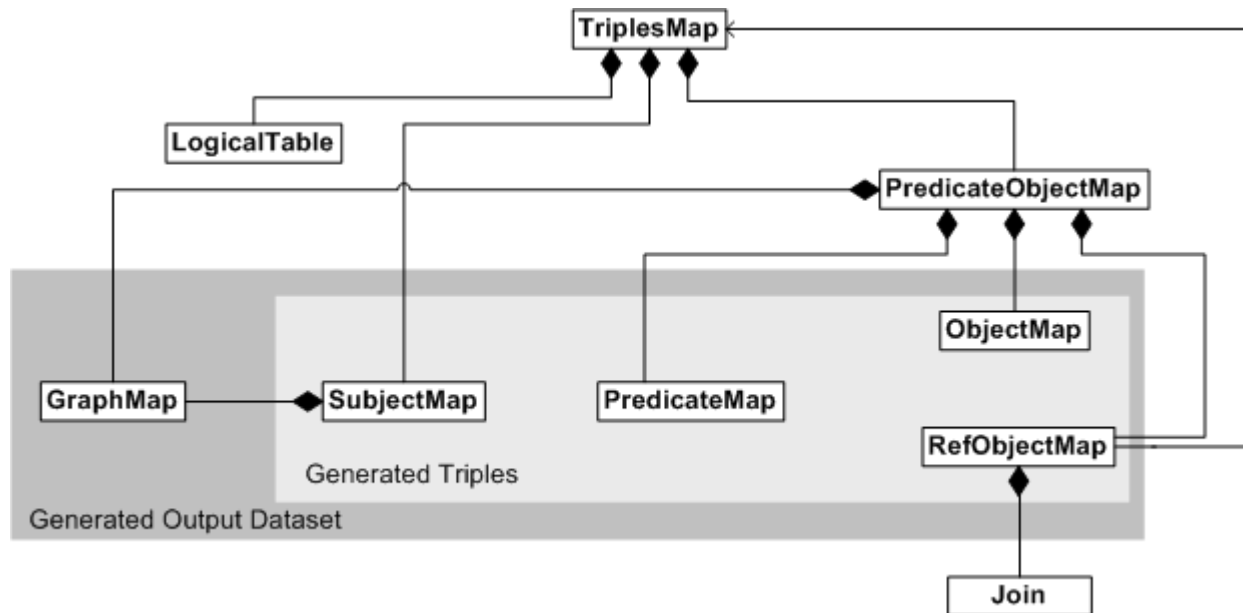
- O *SubjectMap* cria o sujeito das triplas RDF geradas a partir das tuplas da tabela lógica.
- Geralmente é uma *IRI* gerada a partir das colunas que são chave primária da tabela.
- Um *PredicateObjectMap* é composto de um *PredicateMap* e de um *ObjectMap*.

# R2RML

- As triplas são produzidas pela combinação de um *SubjectMap*, um *PredicateMap* e de um *ObjectMap*.

```
<#BookTriplesMap>
  rr:logicalTable [ rr:tableName "book" ];
  rr:subjectMap [
    rr:template "book/{title}";
    rr:class sales:Book;
  ];
  rr:predicateObjectMap [
    rr:predicate sales:title;
    rr:objectMap [ rr:column "title" ];
  ];
```

# R2RML



# Referências

1. Resource Description Framework (RDF) Model and Syntax Specification, W3C Proposed Recommendation 05 January 1999. Disponível em <http://www.w3.org/TR/PR-rdf-syntax/>. Acessado em 2012.
2. Tim Berners-Lee: Linked Data, Design Issues. Disponível em: <http://www.w3.org/DesignIssues/LinkedData.html> >. Acessado em 2012.
3. A Direct Mapping of Relational Data to RDF, Marcelo Arenas, Eric Prud'hommeaux, Juan Sequeda, Editors. World Wide Web Consortium, 24 de Março de 2011. Disponível em <http://www.w3.org/TR/rdb-direct-mapping/>.

# Referências

4. Percy E. Salas, Karin K. Breitman, Marco A. Casanova, José Viterb: StdTrip: Promoting the Reuse of Standard Vocabularies in Open Government Data.
5. David Beckett, Tim Berners-Lee: Turtle - Terse RDF Triple Language. Disponível em <http://www.w3.org/TeamSubmission/2008/SUBM-turtle-20080114/> .
6. M. Dürst and M. Suignard: Internationalized Resource Identifiers (IRIs), Internet-Draft, June 2003, expires December 2003. This document is <http://www.w3.org/International/iri-edit/draft-duerst-iri-04>.



# Referências

7. Das Souripriya, Sundara Seema, Cyganiak Richard (2011), R2RML: RDB to RDF Mapping Language. Disponível em <<http://www.w3.org/TR/r2rml/>>
8. Bizer, C., Cyganiak, R.: D2R server – publishing relational databases on the Semantic Web. Disponível em:  
<http://www4.wiwiiss.fu-berlin.de/bizer/pub/Bizer-Cyganiak-D2R-Server-ISWC2>
9. Jena - A Semantic Web Framework for Java. Disponível em:  
<<http://jena.sourceforge.net>>