

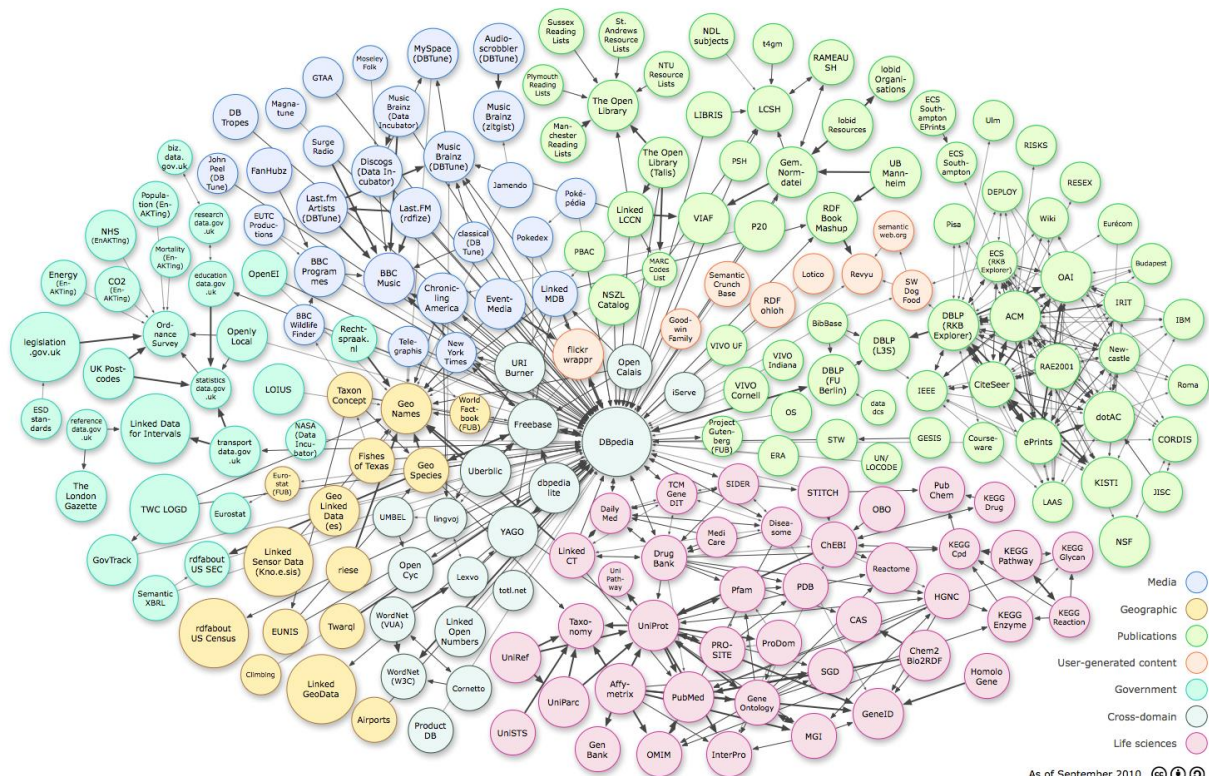
Napredni modeli i baze podataka

Predavanja
Siječanj 2016.

12. Semantičke web tehnologije - SPARQL

Semantički web – mreža podataka

- Web
- Semantički web
- Semantičke web tehnologije
 - RDF
 - RDFS
 - **SPARQL**
 - Ontologije
- Mreža podataka



100

Kako dohvatiti neke podatke iz RDF/RDFS-a?

Kako pretraživati graf?

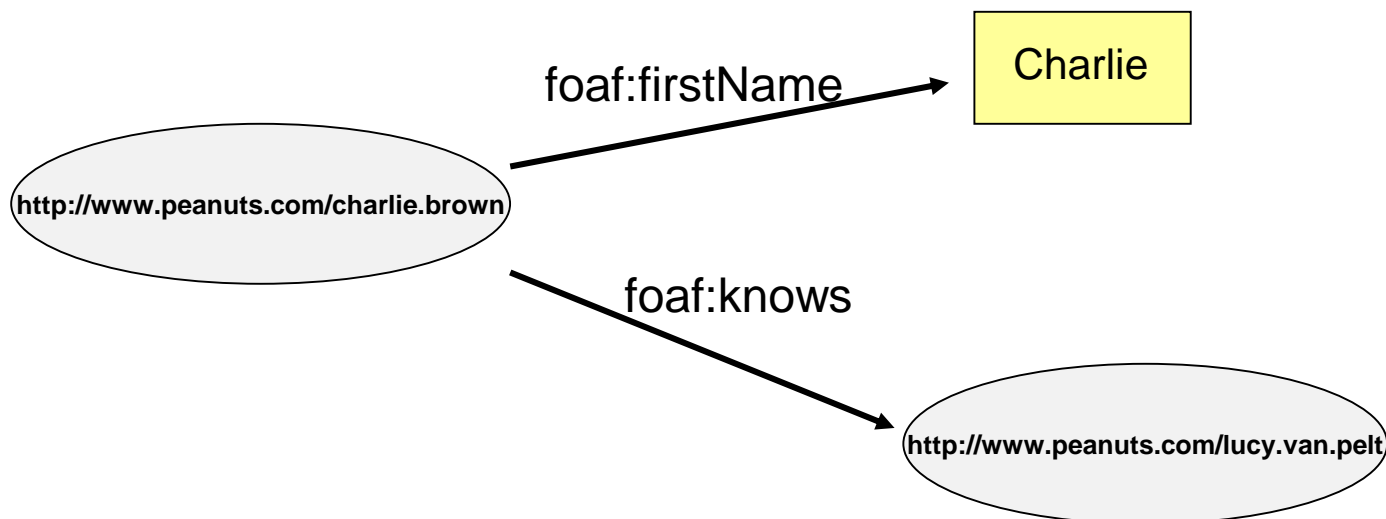
Kako dohvatiti podgraf?



Dohvat podataka iz RDF-a – “naivni” pristupi

- Ručno čitanje ili parsiranje podataka pohranjenih u RDF trojkama (npr. parsiranje RDF/XML formata s nekim XML parserom)
- Korištenje nekog od postojećih RDF API-ja i metoda koje taj API nudi.

```
$personURI = "http://www.peanuts.com/charlie.brown";  
$resource = $graph->resource( $personURI );  
print $resource->get( "foaf:firstName" );
```



Dohvat podataka iz RDF-a – “naivni” pristupi

```
String resource = "http://www.peanuts.com/charlie.brown";
Resource r = newModel.getResource(resource);

StmtIterator iter = r.listProperties();

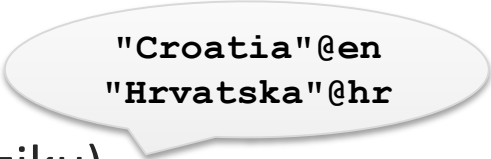
while (iter.hasNext()) {
    Property predicate;
    RDFNode object;
    Statement stmt = iter.nextStatement();

    predicate = stmt.getPredicate();
    object = stmt.getObject();

    System.out.println(predicate + ":" + object);
}
```

Dohvat podataka iz RDF-a

- RDF je grafovska “baza podataka”.
- Podatke iz baze podataka dohvaćamo pomoću SQL-a.
- TREBA NAM:
 - Alat sličan SQL-u (slične sintakse)
 - S “naprednim” mogućnostima na koje smo navikli u SQL-u (grupiranje, agregatne funkcije...)
 - Prilagođen strukturi grafa
 - Prilagođen specifičnostima RDF-a (npr. inf. o jeziku)
 - Prilagođen specifičnostima “on-line” baza podataka



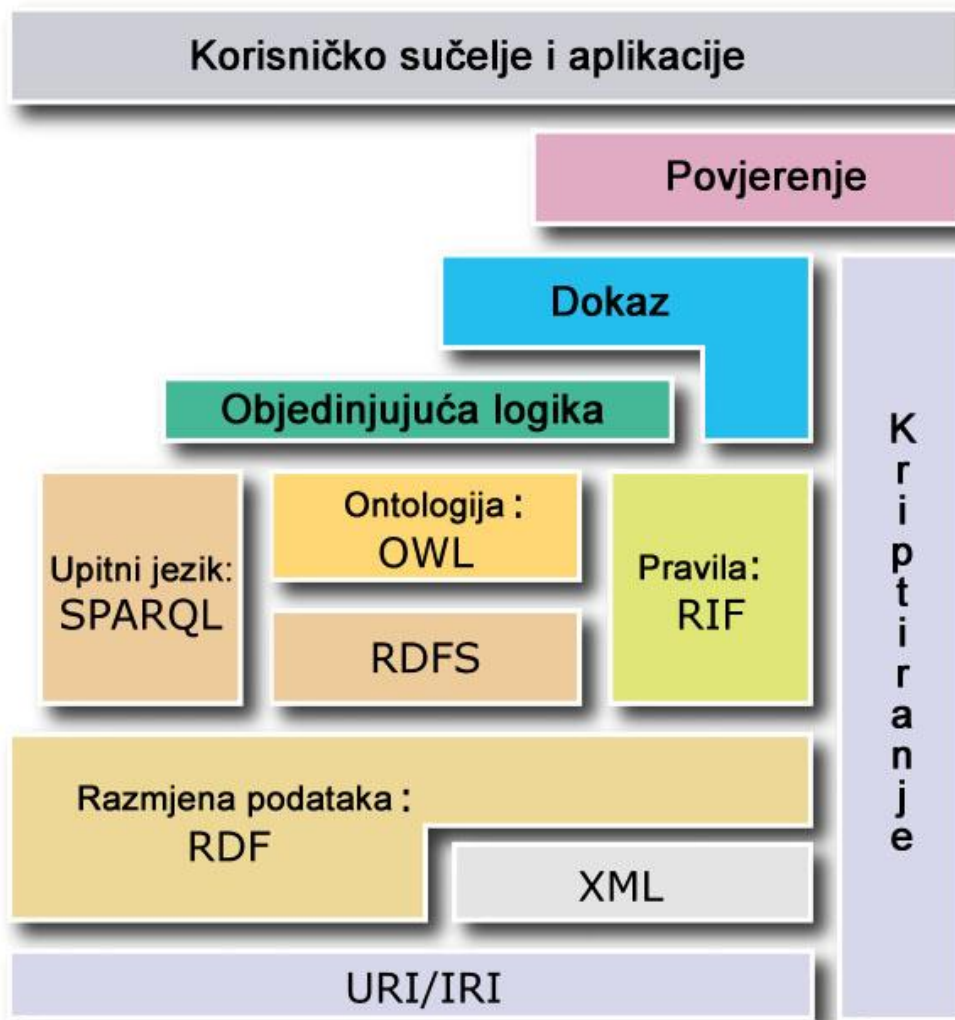
"Croatia"@en
"Hrvatska"@hr

SPARQL = **SPARQL** Protocol and **RDF** Query Language

- upitni jezik za RDF
 - protokol koji definira upotrebu SPARQL upitnog jezika preko http-a
 - specifikacija XML formata za ispis rezultata SPARQL upita
-
- napravljen po uzoru na SQL
 - uvažava i koristi specifičnosti strukture i načina pohrane podataka u RDF-u, te prijenosa podataka http-om
 - W3C standard (*W3C Recommendation*)
 - trenutno u verziji 1.1 (21.3.2013.)
 - još uvijek je vrlo raširena verzija 1.0 (iz 2008)



SPARQL – Semantic Web Layer cake



SPARQL (2)

- zasniva se na usporedbi grafova
- Omogućava:
 - Dohvat podataka u obliku URI-ja, literala, praznih čvorova, podgrafa
 - Projekciju, podupite, agregatne funkcije, negaciju...
 - Pretraživanje grafa u potrazi za nepoznatim relacijama
 - Konstrukciju novog grafa na temelju dohvaćenih podataka
 - Transformacija RDF podataka iz jednog vokabulara u drugi
 - JOIN operacije nad podacima iz heterogenih baza i s raznih izvora (tzv. "Federated Queries"), neovisno da li se podaci nalaze u RDF-u ili im se pristupa kroz neki middleware (npr. alat koji mapira podatke iz relacijske baze podataka u RDF)



SPARQL (3)

- 4 tipa upita za dohvat(čitanje) podataka:
 - SELECT
 - ASK
 - DESCRIBE
 - CONSTRUCT
- SPARQL 1.1
 - Mogućnost izmjene podataka u grafu - UPDATE



SPARQL (4)

- **SELECT** - iz izvora podataka izdvaja one elemente koji se podudaraju sa zadanim uzorcima
- **ASK** - vraća samo logičku vrijednost istine ili laži (*true* ili *false*) u ovisnosti postoje li elementi koji se podudaraju po zadanim uzorcima u upitu
- **CONSTRUCT** - stvara RDF graf prema predlošku zadanom u upitu te koristi *where* dio upita da bi zamijenio varijable u predlošku s konkretnim vrijednostima
- **DESCRIBE** – vraća jedan RDF graf s podacima o URI-u; URI može biti konstanta ili varijabla čija se vrijednost dobije iz *where* dijela upita



SPARQL – SELECT upit

- SPARQL SELECT upit ima 5 tipičnih dijelova
 - Definicija prefiksa ("prostora imena")
 - **PREFIX foaf: <http://xmlns.com/foaf/0.1/>**
 - Definicija dataset-a (grafa ili modela nad kojim radimo upit)
 - **FROM ...**
 - Ako upite postavljamo nad „default” grafom, može se izostaviti
 - Tzv. "Result clause" tj. "SELECT lista" (što želimo dohvatiti)
 - **SELECT ...**
 - Lista Uzoraka koje trojke moraju zadovoljiti
 - **WHERE { ... }**
 - Dodatni modifikatori upita
 - **ORDER BY ...**



SPARQL – SELECT upit - varijable

- definiramo "uzorak" RDF trojki koje želimo dohvatiti
- bilo koji element trojke možemo zamijeniti **varijablom**
- npr.

S

P

O

<http://www.peanuts.com/charlie.brown> <foaf:firstName> "Charlie"

<http://www.peanuts.com/charlie.brown> <foaf:firstName> ?ime

<http://www.peanuts.com/charlie.brown> ?predikat "Charlie"

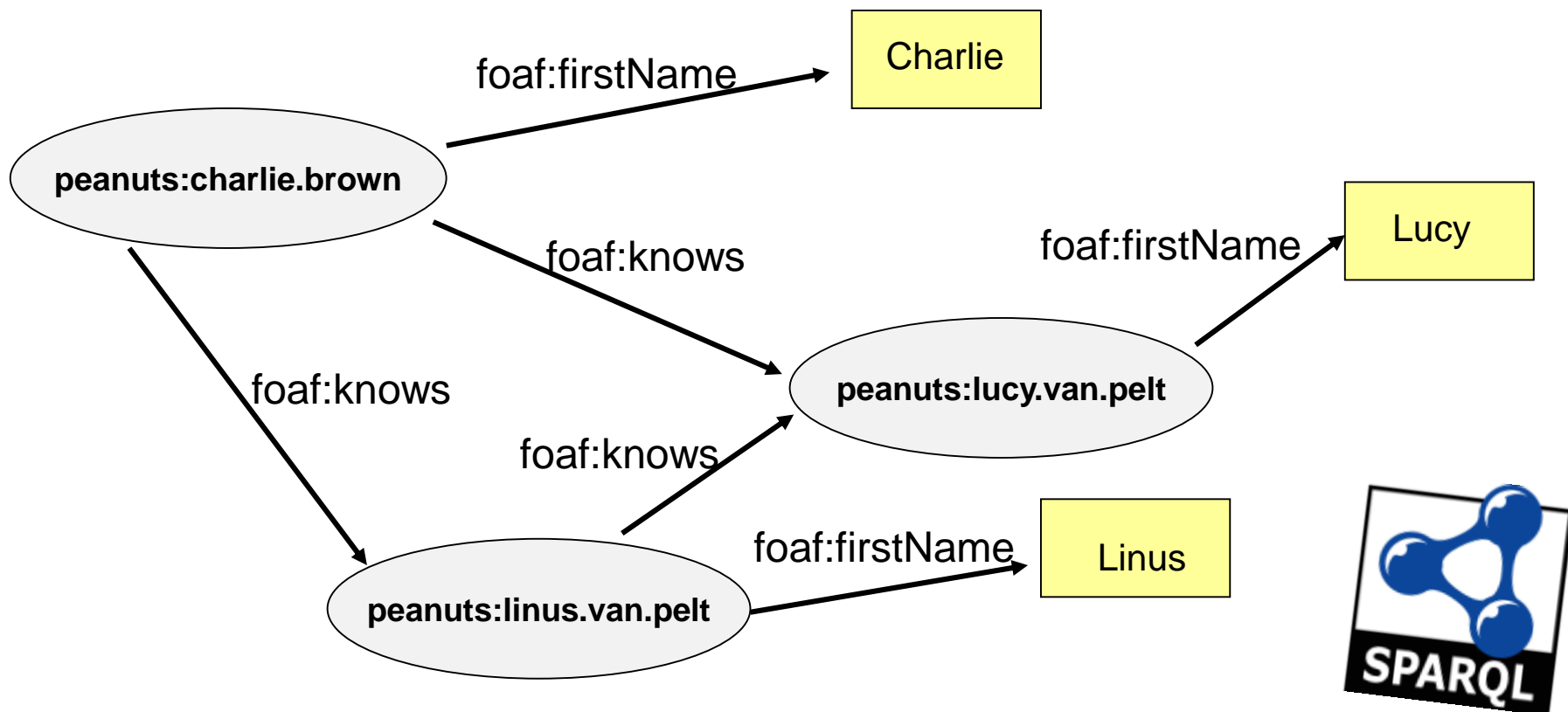
?subjekt ?predikat "Charlie"



SPARQL – SELECT upit - uzorci

- Tražimo sve trojke koje zadovoljavaju ovaj uzorak:
 - (Tražimo sve likove iz Peanutsa i njihova imena)

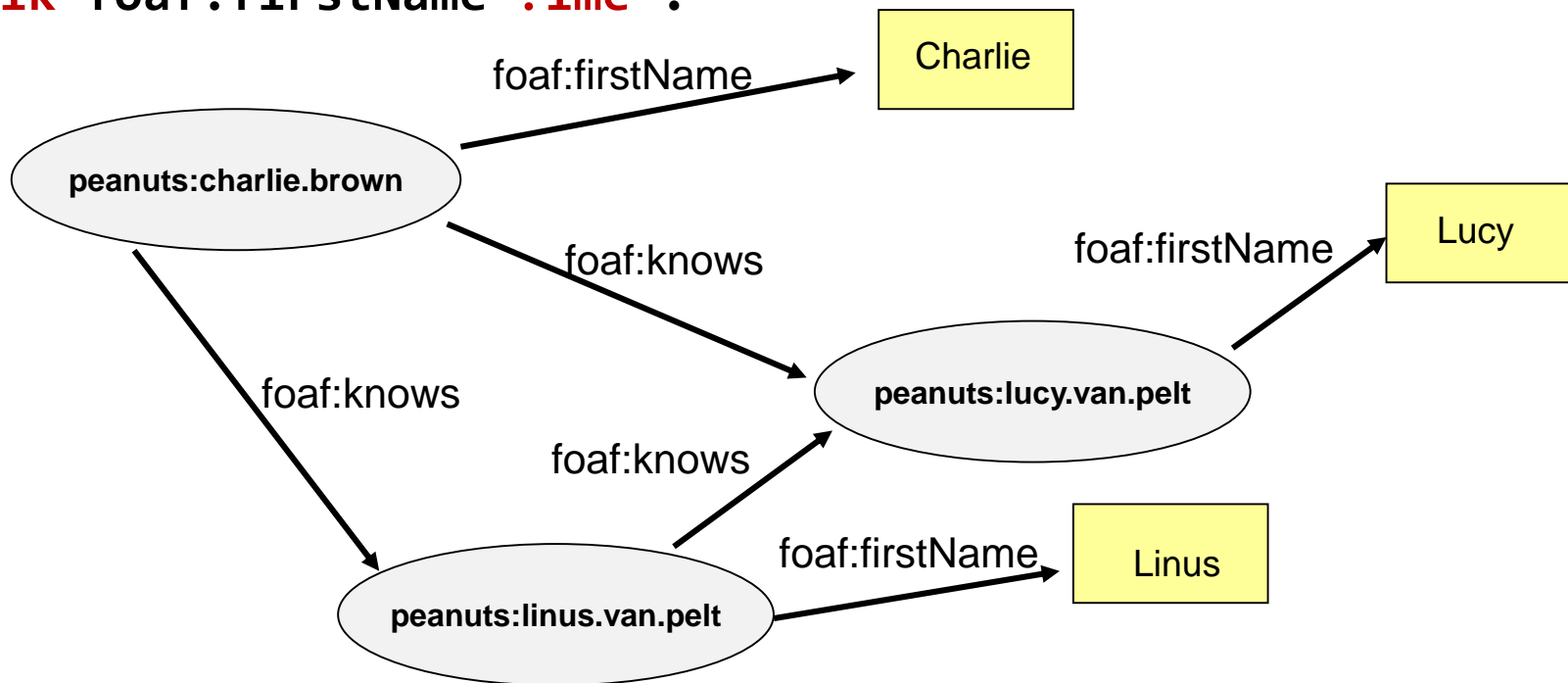
?lik foaf:firstName **?ime** .



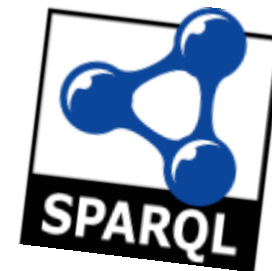
SPARQL – SELECT upit - uзорci

- Uzorak:

?lik foaf:firstName **?ime** .



?lik	?ime
peanuts:charlie.brown	Charlie
peanuts:lucy.van.pelt	Lucy
peanuts:linus.van.pelt	Linus



SPARQL – SELECT upit - uзорci

- Uzorak:

?lik foaf:firstName **?ime** .

- RDF graf:

peanuts:charlie.brown foaf:firstName "Charlie" .
peanuts:charlie.brown foaf:knows peanuts:lucy.van.pelt .
peanuts:charlie.brown foaf:knows peanuts:linus.van.pelt .
peanuts:lucy.van.pelt foaf:firstName "Lucy" .
peanuts:linus.van.pelt foaf:firstName "Linus" .
peanuts:linus.van.pelt foaf:knows peanuts:lucy.van.pelt .

?lik	?ime
peanuts:charlie.brown	Charlie
peanuts:lucy.van.pelt	Lucy
peanuts:linus.van.pelt	Linus

SPARQL – SELECT upit - uзорci

- Uzorak:

?lik foaf:firstName **?ime** .

?lik foaf:knows peanuts:lucy.van.pelt .

- RDF graf:

peanuts:charlie.brown foaf:firstName "Charlie" .

peanuts:charlie.brown foaf:knows peanuts:lucy.van.pelt .

peanuts:charlie.brown foaf:knows peanuts:linus.van.pelt .

peanuts:lucy.van.pelt foaf:firstName "Lucy" .

peanuts:linus.van.pelt foaf:firstName "Linus" .

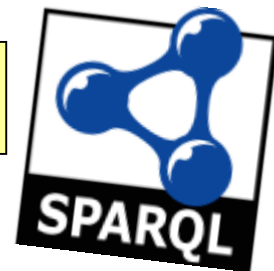
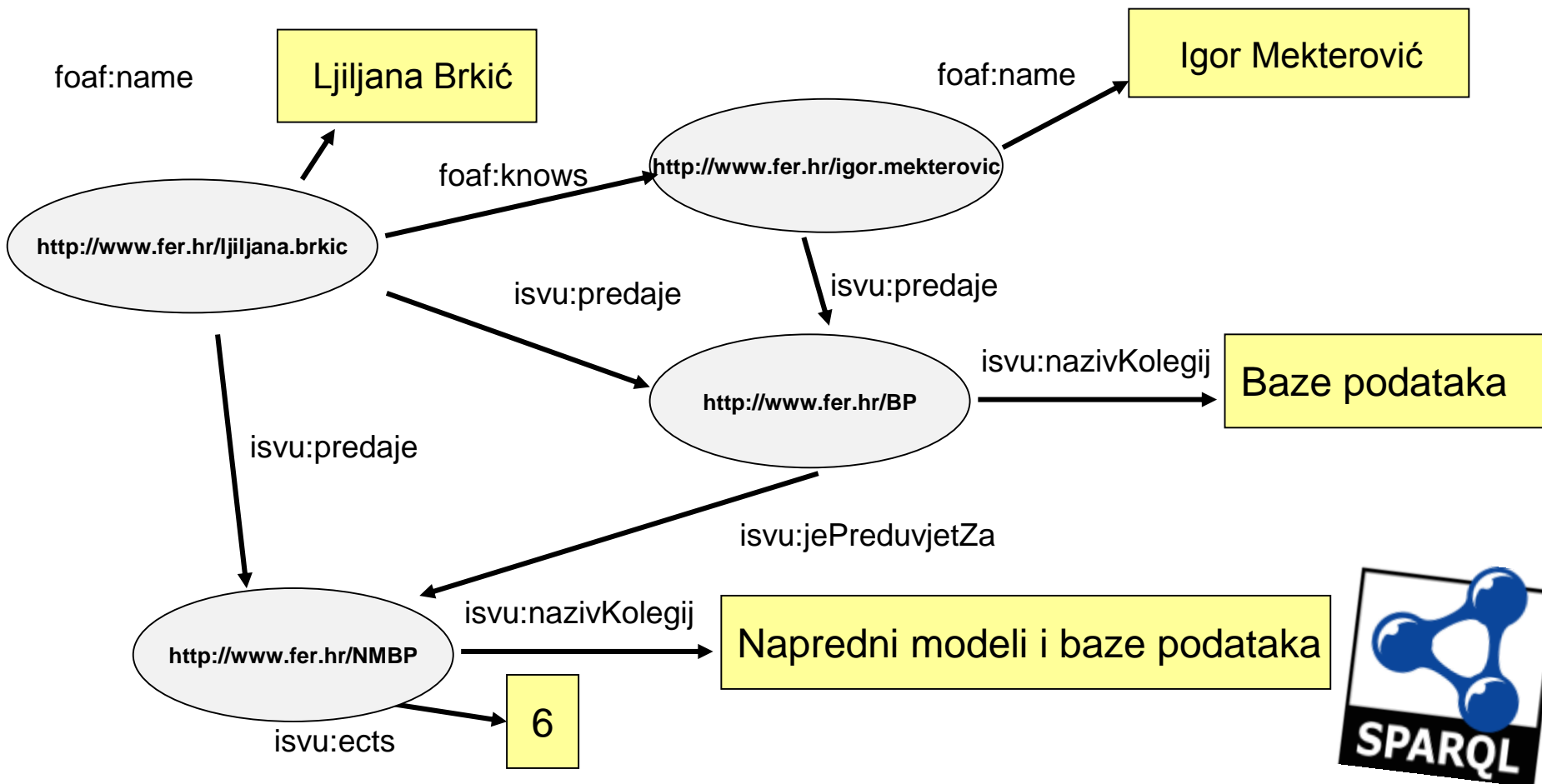
peanuts:linus.van.pelt foaf:knows peanuts:lucy.van.pelt .

?lik	?ime
peanuts:charlie.brown	Charlie
peanuts:linus.van.pelt	Linus

SPARQL – SELECT upit - uzorci

- Koji predmet je preduvjet za upis NMBP?

?predmet isvu:jePreduvjetZa <http://www.fer.hr/NMBP> .



SPARQL – SELECT upit - uзорci

- Koji predmet je preduvjet za upis NMBP?

?predmet isvu:jePreduvjetZa <http://www.fer.hr/NMBP> .

?predmet
<http://www.fer.hr/BP>

- A ako nas zanima naziv predmeta?

?predmet isvu:jePreduvjetZa <http://www.fer.hr/NMBP> .

?predmet isvu:nazivKolegij **?naziv** .

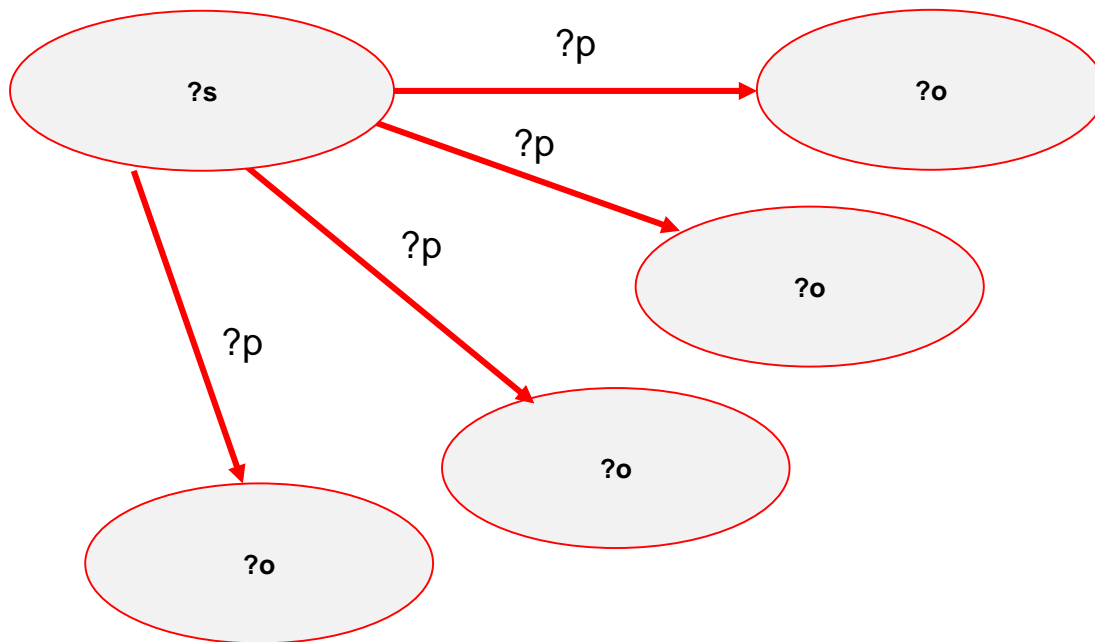
?predmet	?naziv
<http://www.fer.hr/BP>	Baze podataka



SPARQL – SELECT upit - princip (1)

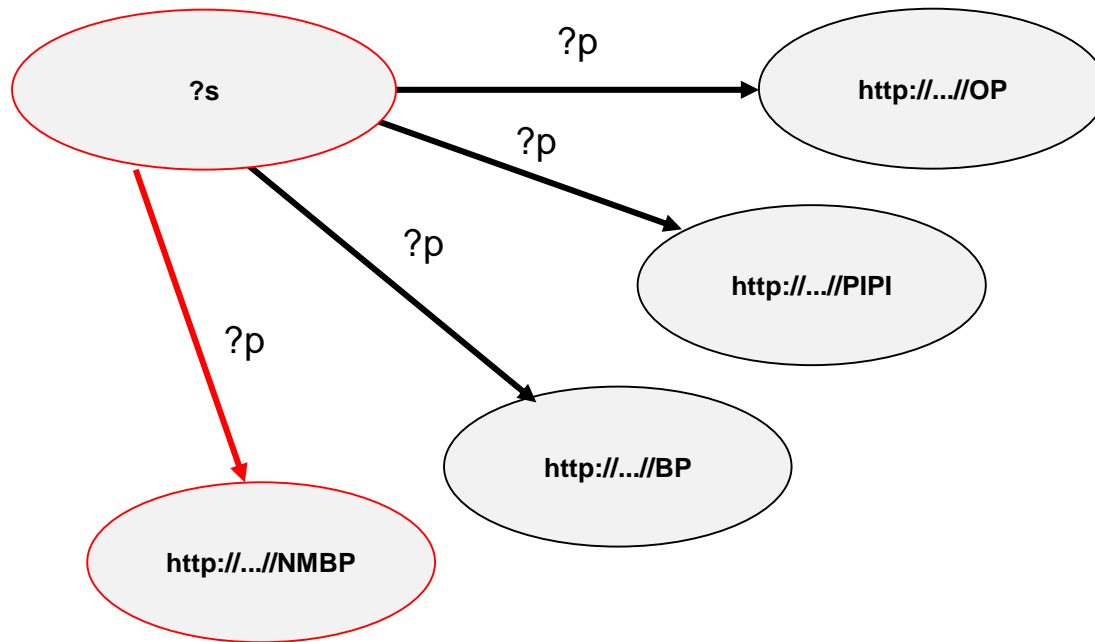
```
SELECT ?p ?o  
WHERE {?s ?p ?o}
```

- Upit vraća sve parove ?p ?o koji zadovoljavaju uzorak grafa definiran pod WHERE dijelom upita; *u gornjem slučaju kao rezultat upita vraća sve parove ?p ?o*



SPARQL – SELECT upit - princip (2)

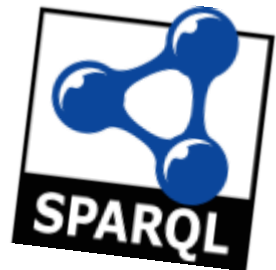
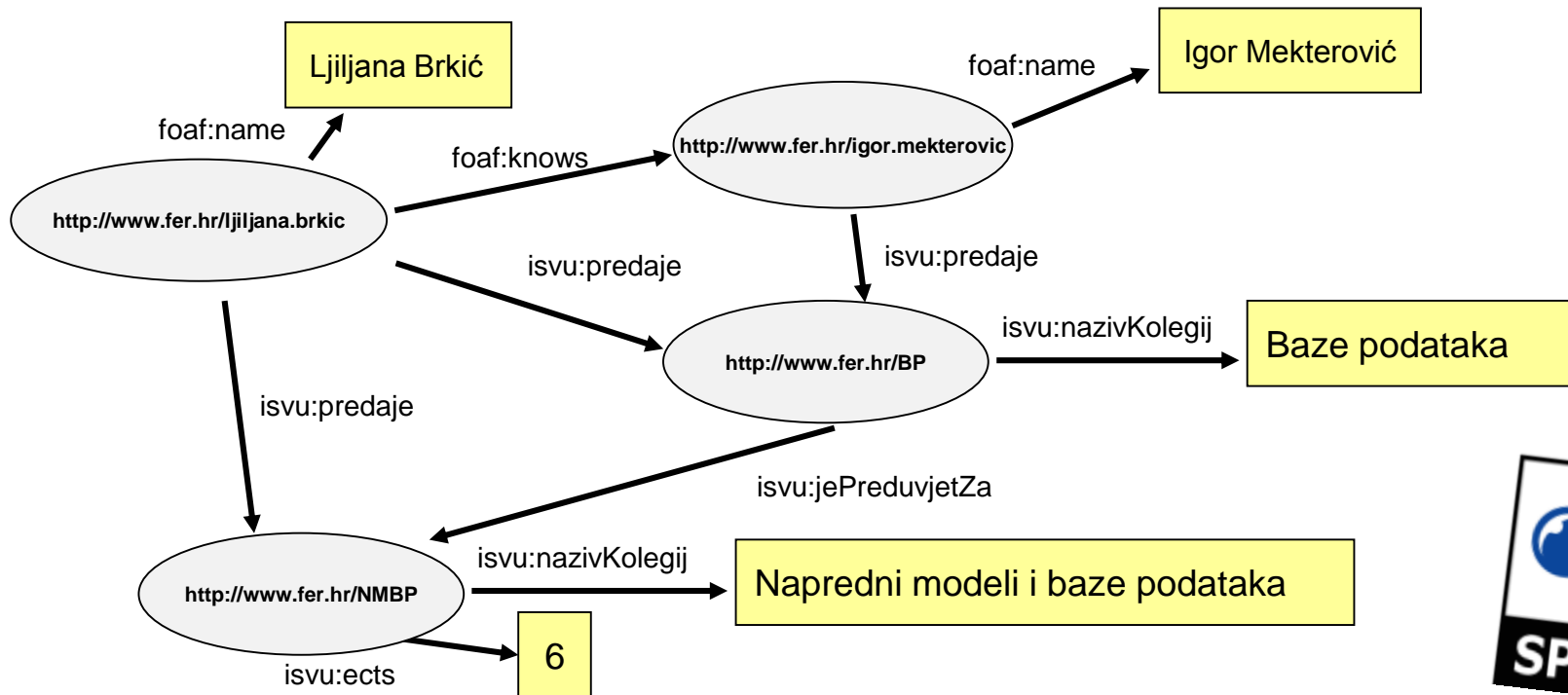
```
SELECT ?s ?p
WHERE {
    ?s ?p <http://www.fer.hr/NMBP>
}
```



SPARQL – SELECT upit – primjer (1)

- Koji predmet je preduvjet za upis NMBP?

```
SELECT ?predmet ?naziv
WHERE {
  ?predmet isvu:jePreduvjetZa <http://www.fer.hr/NMBP> .
  ?predmet isvu:nazivKolegij ?naziv .
}
```



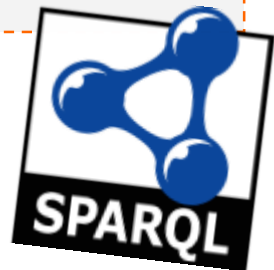
SPARQL – SELECT upit – primjer (1)

```
SELECT ?predmet ?naziv
WHERE {
    ?predmet isvu:jePreduvjetZa <http://www.fer.hr/NMBP> .
    ?predmet isvu:nazivKolegij ?naziv .
}
```



- Kako znamo što je "isvu:jePreduvjetZa"?
- Treba nam definicija prostora imena, tj. **PREFIX**

```
PREFIX isvu: <http://www.isvu.hr/sw/isvu-rdfs#>
SELECT ?predmet ?naziv
WHERE {
    ?predmet isvu:jePreduvjetZa <http://www.fer.hr/NMBP> .
    ?predmet isvu:nazivKolegij ?naziv .
}
```



SPARQL – SELECT upit – primjer (2)

- Ne zanima nas URI predmeta nego samo naziv

```
PREFIX isvu: <http://www.isvu.hr/sw/isvu-rdfs#>
SELECT ?naziv
WHERE {
    ?predmet isvu:jePreduvjetZa <http://www.fer.hr/NMBP> .
    ?predmet isvu:nazivKolegij ?naziv .
}
```

- Kako se zovu predavači predmeta koji su preduvjet NMBP-u?

```
PREFIX isvu: <http://www.isvu.hr/sw/isvu-rdfs#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?ime
WHERE {
    ?predmet isvu:jePreduvjetZa <http://www.fer.hr/NMBP> .
    ?osoba isvu:predaje ?predmet .
    ?osoba foaf:name ?ime .
}
```

?ime
Ljiljana Brkić
Igor Mekterović



SPARQL – SELECT upit – ORDER BY

- slično kao u SQL-u

```
ORDER BY <ASC|DESC> (?var)  
LIMIT n  
OFFSET m
```

- korisno za npr. „paging” u pregledu nekih podataka

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
SELECT ?uri ?ime  
WHERE {  
    ?uri foaf:firstName ?ime  
}  
ORDER BY DESC(?ime)
```

?uri	?ime
http://peanuts.com/charlie.brown	Charlie
http://peanuts.com/linus.van.pelt	Linus
http://peanuts.com/lucy.van.pelt	Lucy



WHERE dio koji smo do sada vidjeli služi samo za dohvat trojki koje zadovoljavaju zadani uzorak.

Kako od dohvaćenih trojki izabrati samo one koje imaju neku odgovarajuću vrijednost?

Npr. broj ECTS-a na predmetu je 4?

Tekst je na određenom jeziku?

Tekst je sličan zadanom tekstu?



SPARQL – SELECT upit – FILTER

- FILTER (<IZRAZ>)
- <IZRAZ> može sadržavati varijable i operatore, ili funkcije

```
FILTER (?rezultat > 100)
```

```
FILTER (?starost >= 7 && ?starost <= 77)
```

```
FILTER (?zanr != "romkom" && ?zanr != "ljubavna drama")
```

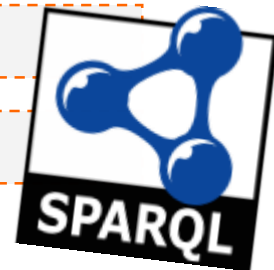
```
FILTER (!BOUND(?datumSmrti))
```

```
FILTER (LANG(?opis) = "EN")
```

```
FILTER regex(?name, "Smith")
```

```
FILTER regex(?name, "smith", "i")
```

```
FILTER regex(STR(?email), "@fer.hr")
```



SPARQL – SELECT upit – FILTER

```
PREFIX isvu: <http://www.isvu.hr/sw/isvu-rdfs#>
SELECT ?naziv ?brECTS
WHERE {
    ?predmet isvu:jePreduvjetZa <http://www.fer.hr/NMBP> .
    ?predmet isvu:ects ?brECTS .
    FILTER (?brECTS = 6)
}
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT ?grad ?ime ?kratkiOpis
WHERE {
    ?grad a dbpedia-owl:City .
    ?grad dbpedia-owl:country <http://dbpedia.org/resource/Croatia> .
    ?grad rdfs:label ?ime .
    ?grad dbpedia-owl:abstract ?kratkiOpis
    FILTER(lang(?ime) = "en")
    FILTER(lang(?kratkiOpis) = "en")
}
```



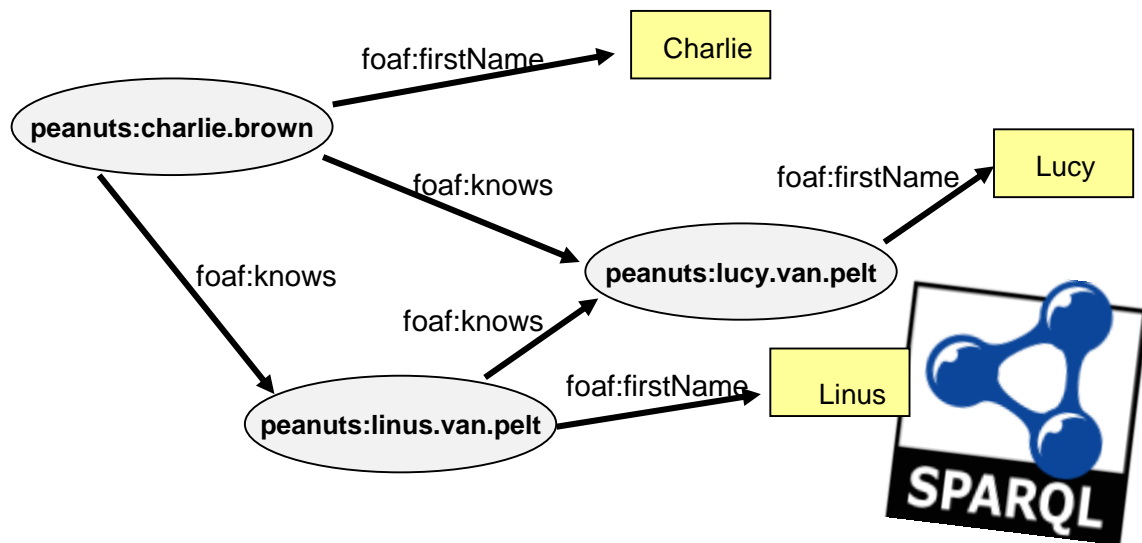
SPARQL – SELECT upit – OPTIONAL

- odgovara LEFT JOIN naredbi u SQL-u

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?ime ?prezime
WHERE {
    ?uri foaf:firstName ?ime
    OPTIONAL { ?uri foaf:familyName ?prezime }
}
```

- veze među uzorcima su tipa „logičko I”

- ovakav upit bez ključne riječi optional ne bi vratio niti jedan zapis



SPARQL – SELECT upit – UNION

- Omogućava uniju, tj. „logičko ILI” među zapisima

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?ime
WHERE {
    { ?uri foaf:firstName ?ime }
    UNION
    { ?uri foaf:name ?ime }
}
```

```
PREFIX imdb: <http://data.linkedimdb.org/resource/movie/>
SELECT DISTINCT ?movie
WHERE {
    { ?movie imdb:actor ?personURI .
      ?personURI imdb:actor_name "Clint Eastwood" }
    UNION
    { ?movie imdb:director ?personURI .
      ?personURI imdb:director_name "Clint Eastwood" }
}
```

SPARQL – ASK upit

- Provjerava postoji li u grafu bar jedan rezultat koji zadovoljava zadani uzorak
- Vraća TRUE ili FALSE

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

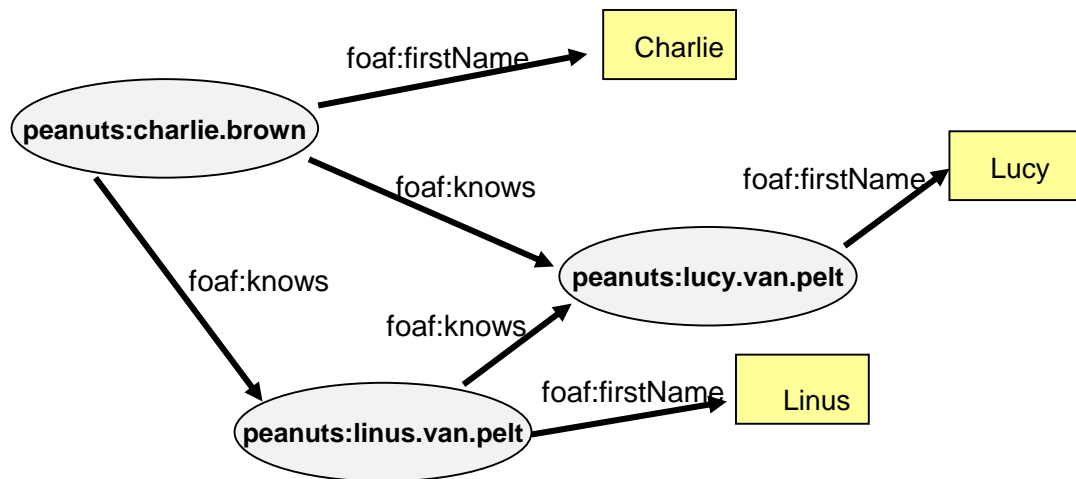
```
ASK
```

```
WHERE {
```

```
  ?uri foaf:firstName "Lucy" .
```

```
  ?uri foaf:knows <http://www.peanuts.com/linus.van.pelt>
```

```
}
```



?result

false

SPARQL – DESCRIBE i CONSTRUCT upiti

- DESCRIBE Vraća RDF graf (u RDF/XML ili Turtle formatu) s podacima o zadanom resursu

```
DESCRIBE <http://www.peanuts.com/linus.van.pelt>

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
DESCRIBE ?r
WHERE {
    ?r foaf:firstName "Lucy"
}
```

- CONSTRUCT definira obrazac po kojem će se u stvoriti novi graf

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
CONSTRUCT { ?r foaf:name ?name }
WHERE {
    ?r foaf:firstName ?name
}
```


SPARQL – upiti – primjeri (1)

SELECT

```
PREFIX foaf:http://xmlns.com/foaf/0.1/  
SELECT DISTINCT ?name ?mbox  
WHERE {  
    ?x foaf:name ?name;  
        foaf:mbox ?mbox}  
ORDER BY DESC(?name)  
LIMIT 5
```

ASK

```
PREFIX foaf:http://xmlns.com/foaf/0.1/  
ASK  
{?s foaf:firstName "Charlie"}
```

DESCRIBE

```
DESCRIBE <http://kent.zpr.fer.hr:8080/educationalProgram/resource/person/100098>
```

Upute možete vježbati na:

<http://dbpedia.org/sparql>

<http://www.linkedmdb.org/snorql/>

<http://kent.zpr.fer.hr:8080/educationalProgram/snorql/>

<http://zpr.fer.hr/linkedata.php>



SPARQL upiti i primjeri (2)

■ One degree from Kevin Bacon

```
PREFIX imdb: <http://data.linkedmdb.org/resource/movie/>
SELECT DISTINCT ?actorName
WHERE {
    ?kb imdb:actor_name "Kevin Bacon" .
    ?movie imdb:actor ?kb .
    ?movie imdb:actor ?actor .
    ?actor imdb:actor_name ?actorName .
    FILTER (?kb != ?actor)
}
ORDER BY ASC(?actorName)
```

```
-----
| actorName                |
=====
| "Aaron Sorkin"           |
| "Adrienne King"         |
| "Aisha Tyler"           |
| "Alec Baldwin"          |
| "Alfre Woodard"         |
| "Alicia Silverstone"    |
| "Alison Lohman"         |
| "Andie MacDowell"       |
| ...                     |
```

SPARQL – SPARQL 1.1

- Agregatne funkcije
 - COUNT, SUM, AVG, MIN, MAX...
- GROUP BY i HAVING
- Podupiti
- EXISTS i NOT EXISTS
- Združeni (en. federated) upiti

SPARQL – SPARQL 1.1 - UPDATE

- Do verzije 1.1 nije bilo moguće dodavati/mijenjati podatke kroz SPARQL
- Puno Web servera još uvijek ne podržava sparql 1.1
 - NEMA „UPDATE” naredbe, tu su:
 - INSERT
 - DELETE
 - LOAD
 - CLEAR
 - DROP
 - COPY

```
INSERT DATA {?s ?p ?o}  
  
DELETE DATA {?s ?p ?o}  
  WHERE {?s a foaf:Person}  
  
LOAD <uri>
```

SPARQL – Združeni upiti

- Omogućava izvršavanje dijela upita (podupita) na drugom serveru (en. endpoint)
- ključna riječ SERVICE

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
  <http://peanuts.com/charlie.brown> foaf:knows ?person .
  SERVICE <http://people.example.org/sparql>
  {
    ?person foaf:name ?name .
  }
}
```

- paziti na redoslijed izvršavanja upita!
 - za svaku vrijednost varijable ?person izvrši se jedan upit na "unutarnjem" endpoint-u

SPARQL – Združeni upiti

```
PREFIX imdb: <http://data.linkedmdb.org/resource/movie/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX dbpo: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?dbpediaUrl ?movieTitle ?movieDate ?filmAbstract ?runtime ?wikiPage ?picURL
WHERE {
    ?actor imdb:actor_name "Arnold Schwarzenegger".
    ?movie imdb:actor ?actor .
    ?movie dcterms:title ?movieTitle .
    ?movie dcterms:date ?movieDate .
    ?movie owl:sameAs ?dbpediaUrl .
    FILTER( REGEX( STR(?dbpediaUrl), "dbpedia")) .
    SERVICE <http://dbpedia.org/sparql>
    {
        ?dbpediaUrl a dbpo:Film .
        ?dbpediaUrl foaf:isPrimaryTopicOf ?wikiPage .
        ?dbpediaUrl dbpo:abstract ?filmAbstract .
        ?dbpediaUrl dbpo:runtime ?runtime .
        ?dbpediaUrl foaf:depiction ?picURL .
        FILTER(lang(?filmAbstract ) = "en")
    }
}
```

SPARQL – Združeni upiti

- Za izvršavanje združenih upita potreban je SPARQL endpoint koji „zna“ obaviti raspodjelu upita i udruživanje rezultata, tzv. „mediator endpoint“.
 - Fuseki server (tj. Jena ARQ) može obaviti medijaciju
 - upit s prethodne stranice izvršen na lokalnoj kopiji LMDB podataka će raditi, ako ga se izvršava na LMDB ili Dbpediji, neće ☹
 - Ako niti jedan od servera niti sparql library ne podržavaju federated upite, za medijaciju se može iskoristiti bilo koji Fuseki server.
 - Npr: upit se izvršava na lokalnom Fuseki serveru:

```
SELECT ?a ?b
WHERE { SERVICE <http://remoteserver1.com/sparql> {
    SELECT ?a WHERE { ?s1 ?p1 ?a }
}
    SERVICE <http://remoteserver2.com/sparql> {
    SELECT ?b WHERE { ?b ?p2 ?o2 }
}
}
```

SPARQL protokol

- SPARQL Protocol je jednostavna metoda za postavljanje SPARQL upita i primanje rezultata preko HTTP-a. SPARQL URL se sastoji od tri dijela:

1. URL SPARQL endpoint-a

npr. `http://dbpedia.org/sparql`

2. (Opcionalno, kao dio niza koji čini upit) Imena grafova nad kojima se postavlja upit

npr. `default-graph-uri=http://dbpedia.org`

3. (Kao dio niza koji čini upit) Sam SPARQL upit

npr. `Query=SELECT distinct ?x
WHERE {?x a <http://umbel.org/umbel/sc/Artist> }`

Primjer SPARQL upita:

`http://dbpedia.org/sparql?default-graph-
uri=http%3A%2F%2Fdbpedia.org&query=SELECT+distinct+%
3Fx+WHERE+{%3Fx+a+%3Chttp%3A%2F%2Fumbel.org%2Fumbel%
2Fsc%2FArtist%3E+}`

SPARQL protokol - primjer

```
GET /sparql?default-graph-uri=http://dbpedia.org&query=
SELECT DISTINCT ?winner WHERE {
  <http://dbpedia.org/resource/World_Snooker_Championship_2009>
  <http://dbpedia.org/property/winner>
  ?winner
}
Host: dbpedia.org
```

```
HTTP/1.1 200 OK
Server: Virtuoso/06.00.3124 (Solaris) x86_64-sun-solaris2.10-64 VDB
Connection: close
Date: Mon, 09 Nov 2009 08:03:10 GMT
Accept-Ranges: bytes
X-SPARQL-default-graph: http://dbpedia.org
Content-Type: application/sparql-results+xml; charset=UTF-8

<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="winner"/>
  </head>
  <results distinct="false" ordered="true">
    <result>
      <binding name="winner">
        <uri>http://dbpedia.org/resource/John_Higgins_%28snooker_player%29</uri>
      </binding>
    </result>
  </results>
</sparql>
```

Kako saznati prefiks?

```
PREFIX imdb: <http://data.linkedmdb.org/resource/movie/>  
PREFIX dcterms: <http://purl.org/dc/terms/>  
PREFIX dbpo: <http://dbpedia.org/ontology/>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX owl: <http://www.w3.org/2002/07/owl#>  
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

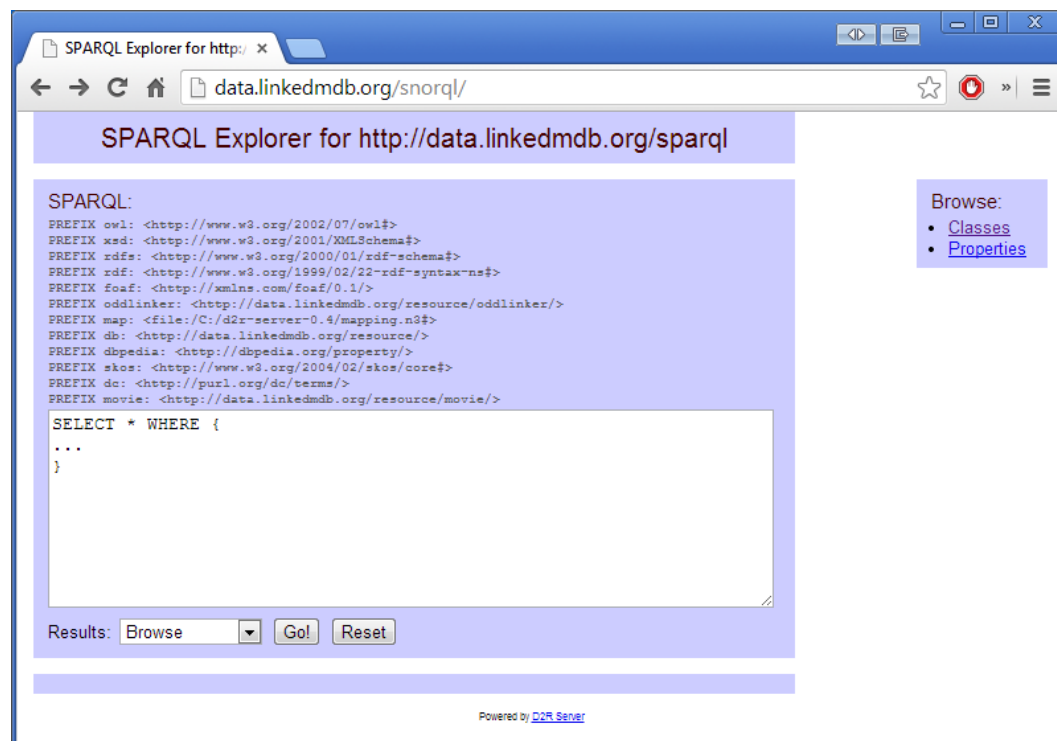
- pogledati u RDF graf
- kroz sučelje za korisnike
 - npr. za Dbpediju: [http://dbpedia.org/page/Predator \(film\)](http://dbpedia.org/page/Predator_(film))
- Uz pomoć neke od specijaliziranih web tražilica
 - <http://prefix.cc/>
- Google 😊

Kako znamo koji sve podaci se nalaze u grafu?

- Ne znamo točno, znamo koji podaci se najčešće nalaze
 - slično i u relacijskim bazama: NULL == nedostajuća trojka
- Možemo dohvatiti popis klasa ili svojstava

```
SELECT DISTINCT ?class  
WHERE { [] a ?class }  
ORDER BY ?class
```

- Kroz web sučelje



Zadatak – projekt iz Semantičkog weba

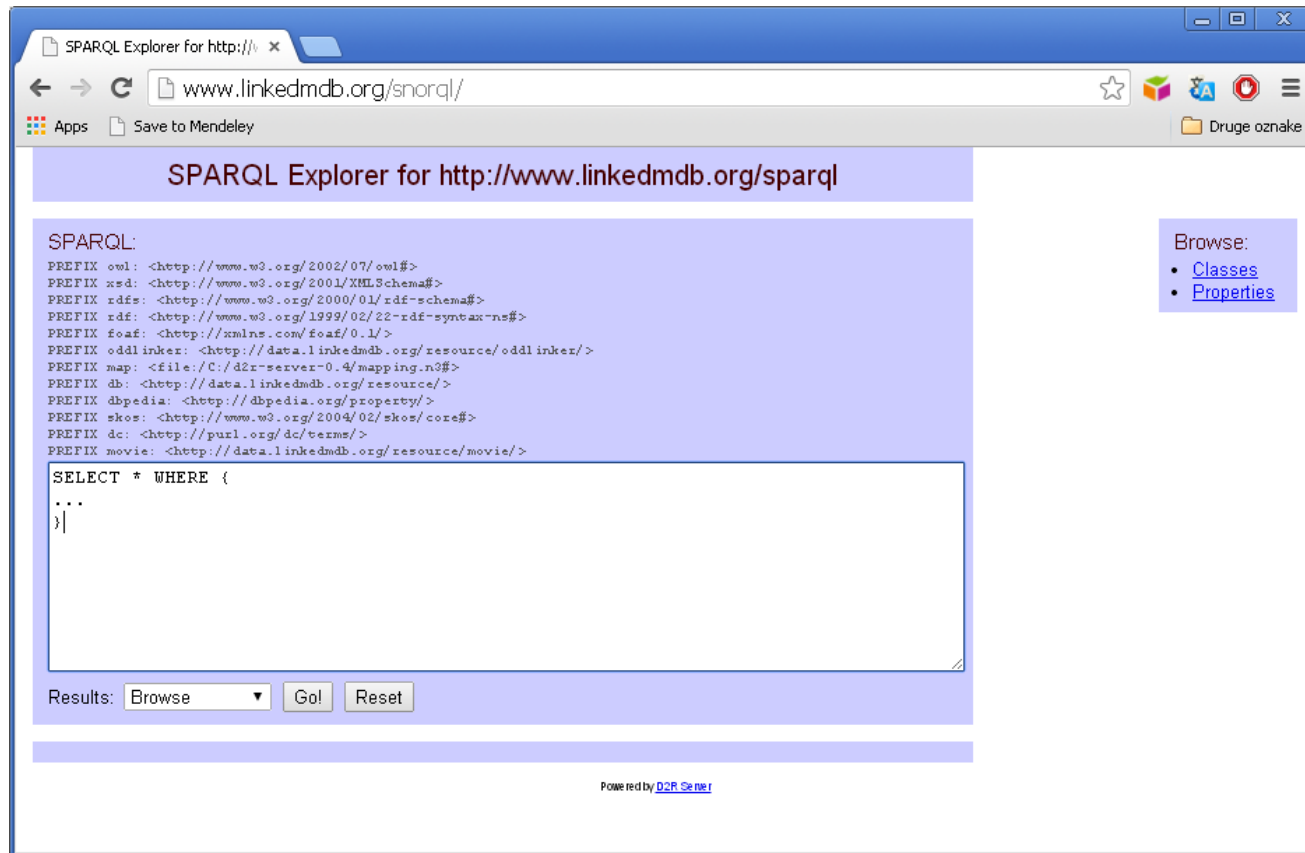
- Potrebno je napraviti jednostavan filmski web portal, koji kao izvor podataka koristi semantičke izvore **Linked movie database** i **Dbpedia**.
- **ROK ZA PREDAJU: (15-19). 2. 2016. u 9:00**
- Rješenje postaviti u vlastiti NM_BP\P4 direktorij na FTP serveru
- Priprema:
 - materijali sa ferweb-a:
 - upute za projekt
 - Apache Jena, Apache Jena Fuseki server, Fuseki+LMDB dump
ili
 - ARC2 library Graphite PHP library, Graphite sqllib library

Zadatak – projekt iz Semantičkog weba

- Priprema
 - Proučiti izvore podataka:
 - <http://dbpedia.org/>, <http://www.linkedmdb.org/>
 - Vježbanje i testiranje upita:
 - <http://dbpedia.org/sparql>, <http://www.linkedmdb.org/snorql/>
 - SPARQL pristupne točke:
 - <http://dbpedia.org/sparql>, <http://data.linkedmdb.org/sparql>

Zadatak – projekt iz Semantičkog weba

- Vježbanje i testiranje upita
 - npr. <http://www.linkedmdb.org/snorql/>



Zadatak – projekt iz Semantičkog weba

■ Vježbanje i testiranje upita

SPARQL Explorer for <http://data.linkedmdb.org/sparql/>

SPARQL:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX oddlinker: <http://data.linkedmdb.org/resource/oddlinker/>
PREFIX map: <file:/C:/d2r-server-0.4/mapping.n3#>
PREFIX db: <http://data.linkedmdb.org/resource/>
PREFIX dbpedia: <http://dbpedia.org/property/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX movie: <http://data.linkedmdb.org/resource/movie/>

PREFIX imdb: <http://data.linkedmdb.org/resource/movie/>
SELECT DISTINCT ?movie (COUNT(?personURI) AS ?ukupnoGlumaca)
WHERE {
    ?movie imdb:actor ?personURI .
}
GROUP BY ?movie
ORDER BY DESC(?ukupnoGlumaca)
LIMIT 10
```

Results:

Powered by [D2R Server](#)

Browse:

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

Zadatak – projekt iz Semantičkog weba

- Lokalna kopija LMDB grafa
 - on-line verzija LMDB-a vraća ograničen podskup podataka
 - on-line verzija može postati nedostupna
 - Fuseki omogućava izvršavanje združenih upita
- **jena-fuseki+dump.zip** raspakirati i pokrenuti Fuseki server:
 - `fuseki-server.bat --loc=LMDB /ds`
 - server se pokreće na lokalnom portu 3030
 - <http://localhost:3030>
 - na linku "control panel" potrebno je izabrati jedini ponuđeni set podataka (dataset) - /ds. Nakon toga otvara se interaktivno sučelje gdje je moguće postavljati upite.
 - SPARQL endpoint: <http://localhost:3030/ds/sparql>

Zadatak – projekt iz Semantičkog weba

- Izvršavanje SPARQL upita iz Jave:
 - u projekt je potrebno uključiti Jena library

```
import com.hp.hpl.jena.query.*;
import com.hp.hpl.jena.rdf.model.*;

public class Test {
    public static void main(String[] args) {

        String ZPR_endpoint = "http://kent.zpr.fer.hr:8080/zprSWeb/sparql";
        String queryString =
            " PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> "
          + " PREFIX foaf: <http://xmlns.com/foaf/0.1/> "
          + " SELECT DISTINCT ?person ?firstName ?surname ?title "
          + " { "
          + "   ?person rdf:type foaf:Person ."
          + "   ?person foaf:firstName ?firstName ."
          + "   ?person foaf:surname ?surname ."
          + "   ?person foaf:title ?title ."
          + " } "
          + " ORDER BY ASC(?surname) ASC(?firstName)";

        Query query = QueryFactory.create(queryString);
        QueryExecution qe = QueryExecutionFactory.sparqlService(ZPR_endpoint, query);

        ResultSet res = qe.execSelect();
        ResultSetFormatter.out(System.out, res);

    }
}
```

Zadatak – projekt iz Semantičkog weba

■ Izvršavanje SPARQL upita iz PHP-a:

```
<?php
require_once( "sparqllib.php" );

$query =
    'PREFIX lmbd: <http://data.linkedmdb.org/resource/movie/>
    SELECT DISTINCT ?movie (COUNT(?personURI) AS ?ukupnoGlumaca)
    WHERE {
        ?movie lmbd:actor ?personURI .
    }
    GROUP BY ?movie
    ORDER BY DESC(?ukupnoGlumaca)
    LIMIT 10';

$data = sparql_get( "http://localhost:3030/ds/sparql", $query);

if( isset($data)) {
    print "<table class='example_table'><tr>";
    foreach( $data->fields() as $field ) {
        print "<th>$field</th>";
    }
    print "</tr>";
    foreach( $data as $row ) {
        print "<tr>";
        foreach( $data->fields() as $field ) {
            print "<td>$row[$field]</td>";
        }
        print "</tr>";
    }
    print "</table>";
?>
```