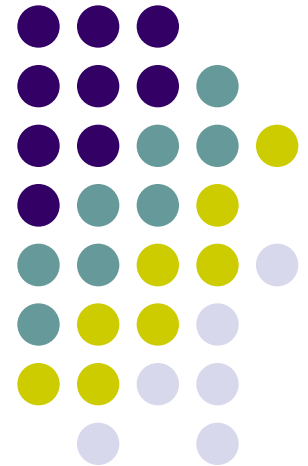
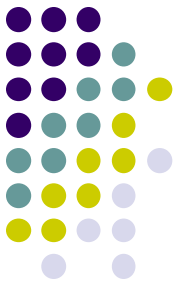


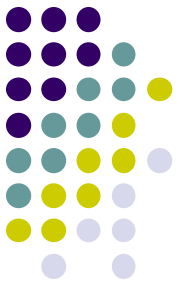
SPARQL





Overview

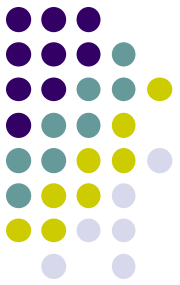
- Introduction
- SPARQL Query Language
 - Graph Patterns
 - Query Ordering
 - Query Forms
 - Testing Values
- SPARQL Protocol and
- SPARQL Results Format
- SPARQL Support



Introduction

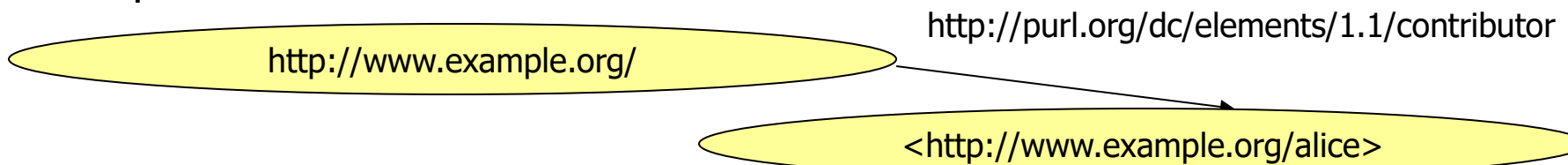
RDF – flexible and extensible way to represent information about WWW resources

SPARQL = SQL-like Query Language for RDF



RDF Notations overview

Graph Notation:

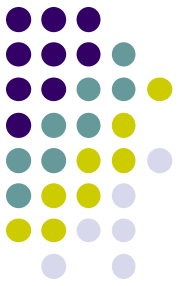


Triple Notation:

```
<http://www.example.org/>  
<http://purl.org/dc/elements/1.1/contributor>  
<http://www.example.org/alice> .
```

RDF/XML Notation:

```
<?xml version="1.0"?>  
  
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"   
          xmlns:dc="http://purl.org/dc/elements/1.1/" >  
  
  <rdf:Description rdf:about="http://www.example.org/" >  
    <dc:contributor rdf:resource="http://www.example.org/alice"/>  
  </rdf:Description>  
  
</rdf:RDF>
```



Turtle notation

- Turtle: An RDF serialization
 - The RDF part of N3
 - Commonly used in examples
 - SPARQL uses Turtle+variables as triple pattern syntax

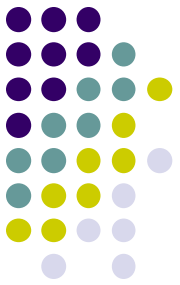
Data description format - Turtle



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

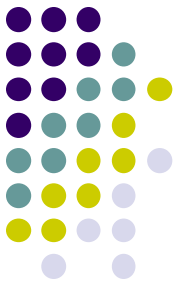
```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ex: <http://example.org/stuff/1.0/> .
```

```
<http://www.w3.org/TR/rdf-syntax-grammar>
  dc:title "RDF/XML Syntax Specification (Revised)" ;
  ex:editor [ ex:fullname "Dave Beckett";
              ex:homePage <http://purl.org/net/dajobe/> ] .
```



SPARQL Query Language

- SPARQL - query language for getting information from RDF graphs. It provides facilities to:
 - 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
- matching graph patterns
- variables – global scope; indicated by ‘?’ or ‘\$’
- query terms – based on Turtle syntax
- terms delimited by "<>" are *relative URI references*
- data description format - Turtle



Simple query

- Data

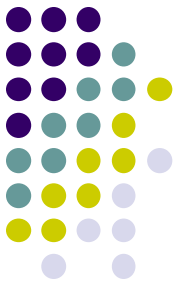
```
<http://example.org/book/book1>  
<http://purl.org/dc/elements/1.1/title> "SPARQL Tutorial" .
```

- Query

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

- Result

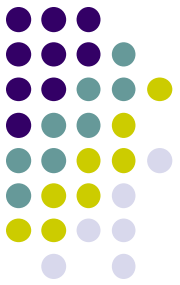
title
"SPARQL Tutorial"



Simple SPARQL example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
FROM <data.rdf>
WHERE { ?x foaf:name ?name }
```

- **SELECT** specifies what the query should return
- **FROM** is an optional clause that provides the URI of the dataset to use.
 - points to local file or the URL of a graph on the Web.
- **WHERE** clause consists of a series of triple patterns, expressed using Turtle-based syntax. These triples together comprise what is known as a *graph pattern*.



Graph Patterns

Basic Graph Pattern – set of *Triple Patterns*

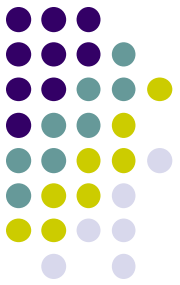
Group Pattern - a set of graph patterns must all match

Value Constraints - restrict RDF terms in a solution

Optional Graph Patterns .- additional patterns may extend the solution

Alternative Graph Pattern – two or more possible patterns are tried

Patterns on Named Graphs - patterns are matched against named graphs



Basic Graph Pattern

- Set of Triple Patterns
 - **Triple Pattern** – similar to an RDF Triple (subject, predicate, object), but any component can be a query variable; literal subjects are allowed

`?book dc:title ?title`

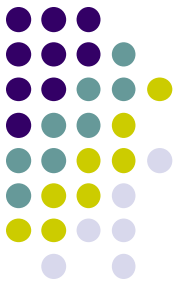
- Matching a triple pattern to a graph: bindings between variables and RDF Terms
- Matching of Basic Graph Patterns
 - A **Pattern Solution** of Graph Pattern GP on graph G is any substitution S such that S(GP) is a subgraph of G.

`SELECT ?x ?v WHERE { ?x ?x ?v }`

x	v
rdf:type	rdf:Property

`rdf:type rdf:type rdf:Property`

Basic Graph Pattern - Multiple Matches

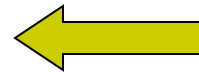


Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Johnny Lee Outlaw" .
_:a foaf:mbox <mailto:jlow@example.com> .
_:b foaf:name "Peter Goodguy" .
_:b foaf:mbox <mailto:peter@example.org> .
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
{ ?x foaf:name ?name .
  ?x foaf:mbox ?mbox }
```

Query



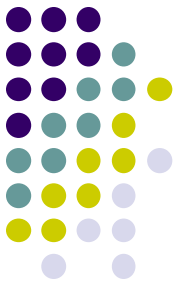
Group Graph Pattern
(set of graph patterns)
also!

Query Result

name	mbox
"Johnny Lee Outlaw"	<mailto:jlow@example.com>
"Peter Goodguy"	<mailto:peter@example.org>

Basic Graph Pattern

-Matching RDF Literals



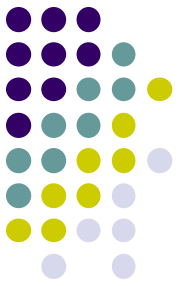
- Data

```
@prefix dt:      <http://example.org/datatype#> .
@prefix ns:      <http://example.org/ns#> .
@prefix :        <http://example.org/ns#> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .

:x      ns:p      "cat"@en .
:y      ns:p      "42"^^xsd:integer .
:z      ns:p      "abc"^^dt:specialDatatype .
```

Basic Graph Pattern

-Matching RDF Literals



- Matching literals with language tags

- Query:

```
SELECT ?v WHERE {?v ?p "cat"}
```

- Result:

v

- Query:

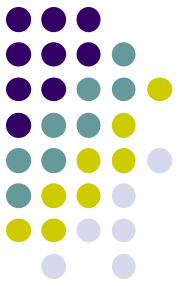
```
SELECT ?v WHERE {?v ?p "cat"@en}
```

- Result:

v
<http://example.org/ns#x>

Basic Graph Pattern

-Matching RDF Literals



- Matching literals with numeric types

- Query:

```
SELECT ?v WHERE {?v ?p 42}
```

- Result:

v
<http://example.org/ns#y>

- Matching literals with arbitrary datatypes

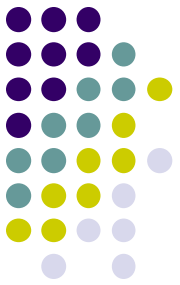
- Query:

```
SELECT ?v WHERE {?v ?p  
"abc"^^<http://example.org/datdatatype#specialDatatype>}
```

- Result:

v
<http://example.org/ns#z>

Basic Graph Pattern - Blank Nodes



- Query results can contain blank nodes.
- Blank nodes are written as “_:” followed by a blank node label
- Data

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

```
_:a      foaf:name      "Alice" .  
_:b      foaf:name      "Bob" .
```

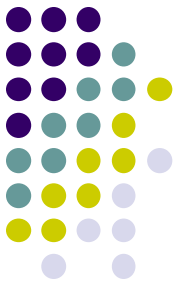
- Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1>  
SELECT ?x ?name  
WHERE {?x foaf:name ?name}
```

- Result

x	name
_:c	"Alice"
_:d	"Bob"

There is no need that the blank node in result should have the same label as in data.



Graph Patterns

Basic Graph Pattern – set of *Triple Patterns*

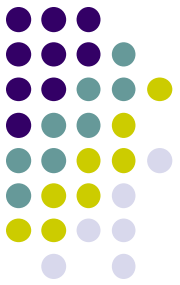
Group Pattern - a set of graph patterns must all match

Value Constraints - restrict RDF terms in a solution

Optional Graph Patterns .- additional patterns may extend the solution

Alternative Graph Pattern – two or more possible patterns are tried

Patterns on Named Graphs - patterns are matched against named graphs



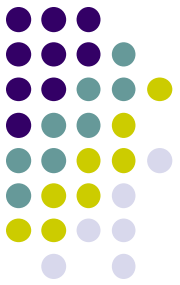
Group Pattern

A set of graph patterns must all match

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
{ ?x foaf:name ?name .
  ?x foaf:mbox ?mbox }
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
{ { ?x foaf:name ?name . }
  { ?x foaf:mbox ?mbox. } }
```

```
PREFIX foaf:
<http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
{ { ?x foaf:name ?name;
  foaf:mbox ?mbox } }
```



Graph Patterns

Basic Graph Pattern – set of *Triple Patterns*

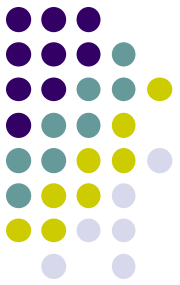
Group Pattern - a set of graph patterns must all match

Value Constraints - restrict RDF terms in a solution

Optional Graph Patterns - additional patterns may extend the solution

Alternative Graph Pattern – two or more possible patterns are tried

Patterns on Named Graphs - patterns are matched against named graphs

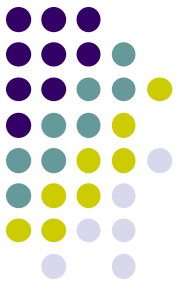


Value Constraints - FILTER

- FILTER can set the constraints for the solutions to those which the filter expression evaluates to be TRUE.
- Data

```
@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 .
```



Value Constraints - FILTER

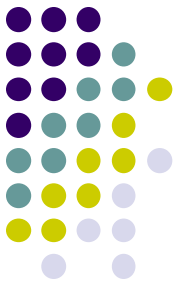
- FILTER functions
 - Can restrict on arithmetic expressions
 - Query

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

SELECT ?title ?price
WHERE { ?x ns:price ?price .
        FILTER (?price < 30.5 )
        ?x dc:title ?title . }
```

- Result

title	price
"The Semantic Web"	23



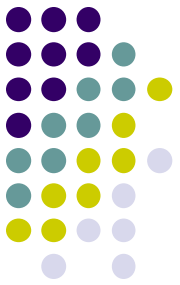
Value Constraints - FILTER

- FILTER functions
 - regex: match only plain literals with no language tag.
 - Query

```
PREFIX dc:      <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { ?x dc:title ?title
        FILTER regex(?title, "^SPARQL")
      }
```

- Result

title
"SPARQL Tutorial"



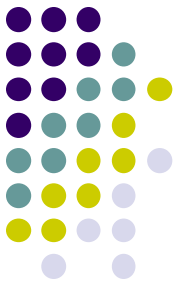
Value Constraints - FILTER

- FILTER functions
 - Regular expression matches may be made case-insensitive with the “i” flag.
 - Query

```
PREFIX dc:      <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { ?x dc:title ?title
        FILTER regex(?title, "web", "i" )
      }
```

- Result

title
"The Semantic Web"



Graph Patterns

Basic Graph Pattern – set of *Triple Patterns*

Group Pattern - a set of graph patterns must all match

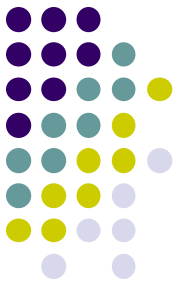
Value Constraints - restrict RDF terms in a solution

Optional Graph Patterns .- additional patterns may extend the solution

Alternative Graph Pattern – two or more possible patterns are tried

Patterns on Named Graphs - patterns are matched against named graphs

Optional graph patterns - Optional matching

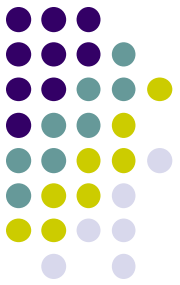


- If optional part does not match, it creates no bindings but does not eliminate the solution.
- Data

```
@PREFIX foaf: <http://xmlns.com/foaf/0.1/> .
@PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

_:a rdf:type foaf:Person .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@example.com> .
_:a foaf:mbox <mailto:alice@work.example> .

_:b rdf:type foaf:person .
_:b foaf:name "Bob" .
```



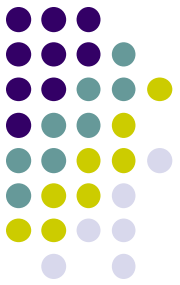
Optional matching

- Query

```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { ?x foaf:name ?name .
        OPTIONAL { ?x foaf:mbox ?mbox }
}
```

- Result

name	mbox
"Alice"	<mailto:alice@example.com>
"Alice"	<mailto:alice@work.example>
"Bob"	

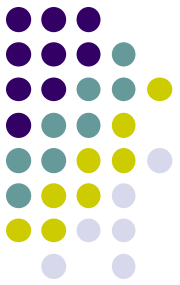


Optional matching

- Data

```
@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 .
```



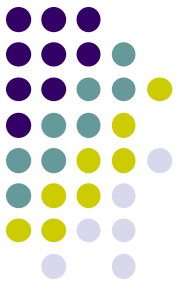
Optional matching

- Query

```
PREFIX dc:      <http://purl.org/dc/elements/1.1/>
PREFIX ns:      <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x dc:title ?title .
       OPTIONAL { ?x ns:price ?price .
                 FILTER (?price < 30) }
}
```

- Result

title	price
"SPARQL Tutorial"	
"The Semantic Web"	23



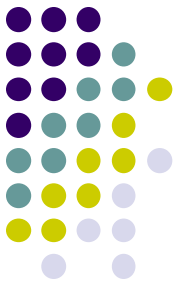
Multiple optional matching

- Data

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

_:a foaf:name          "Alice" .
_:a foaf:homepage      <http://work.example.org/alice/> .

_:b foaf:name          "Bob" .
_:b foaf:mbox          <mailto:bob@work.example> .
```



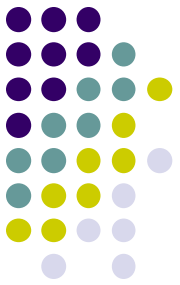
Multiple optional matching

- Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox ?hpage
WHERE { ?x foaf:name ?name .
        OPTIONAL { ?x foaf:mbox ?mbox } .
        OPTIONAL { ?x foaf:homepage ?hpage }
}
```

- Result

name	mbox	hpage
"Alice"		<http://work.example.org/alice/>
"Bob"	<mailto:bob@work.example>	



Graph Patterns

Basic Graph Patterns – set of *Triple Patterns*

Group Patterns - a set of graph patterns must all match

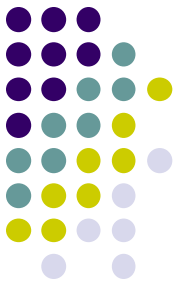
Value Constraints - restrict RDF terms in a solution

Optional Graph Patterns .- additional patterns may extend the solution

Alternative Graph Patterns – two or more possible patterns are tried

Patterns on Named Graphs - patterns are matched against named graphs

Alternative Graph Patterns - Matching Alternatives



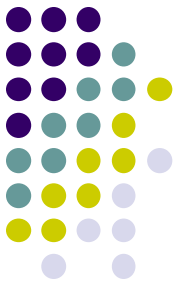
- SPARQL provides a means of combining graph patterns using “UNION”.
- Data

```
@PREFIX dc10: <http://purl.org/dc/elements/1.0/> .
@PREFIX dc11: <http://purl.org/dc/elements/1.1/> .

_:a dc10:title      "SPARQL Query Language Tutorial" .
_:a dc10:creator    "Alice" .

_:b dc11:title      "SPARQL Protocol Tutorial" .
_:b dc11:creator    "Bob" .

_:c dc10:title      "SPARQL" .
_:c dc11:title      "SPARQL (updated)" .
```

Matching Alternatives

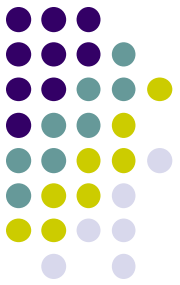
- Query

```
PREFIX dc10: <http://purl.org/dc/elements/1.0/>
PREFIX dc11: <http://purl.org/dc/elements/1.1/>

SELECT ?title
WHERE { { ?book dc10:title ?title } UNION { ?book
dc11:title ?title} }
```

- Result

title
"SPARQL Query Language Tutorial"
"SPARQL"
"SPARQL Protocol Tutorial"
"SPARQL (updated)"



Matching Alternatives

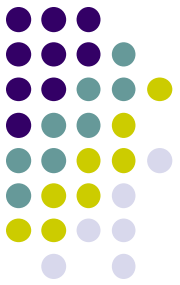
- Query

```
PREFIX dc10: <http://purl.org/dc/elements/1.0/>
PREFIX dc11: <http://purl.org/dc/elements/1.1/>

SELECT ?x ?y
WHERE { { ?book dc10:title ?x } UNION { ?book
dc11:title ?y} }
```

- Result

x	y
"SPARQL Query Language Tutorial"	
"SPARQL"	
	"SPARQL Protocol Tutorial"
	"SPARQL (updated)"



Matching Alternatives

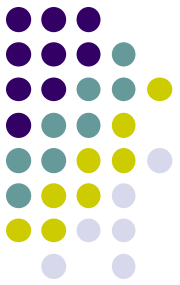
- Query

```
PREFIX dc10: <http://purl.org/dc/elements/1.0/>
PREFIX dc11: <http://purl.org/dc/elements/1.1/>

SELECT ?title ?author
WHERE { { ?book dc10:title ?title . ?book dc10:creator ?author }
        UNION
        { ?book dc11:title ?title . ?book dc11:creator ?author}
      }
```

- Result

title	author
"SPARQL Query Language Tutorial"	"Alice"
"SPARQL Protocol Tutorial"	"Bob"



Graph Patterns

Basic Graph Pattern – set of *Triple Patterns*

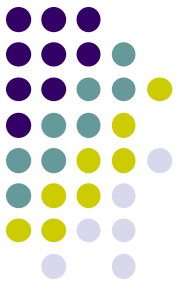
Group Pattern - a set of graph patterns must all match

Value Constraints - restrict RDF terms in a solution

Optional Graph Patterns .- additional patterns may extend the solution

Alternative Graph Pattern – two or more possible patterns are tried

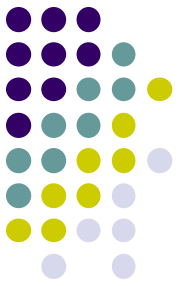
Patterns on Named Graphs - patterns are matched against named graphs



Specifying RDF datasets

- An RDF Dataset comprises
 - one graph, the default graph, which does not have a name, and
 - zero or more named graphs, where each named graph can be identified by an URIs.
- A SPARQL query may specify the dataset to be used for matching by using the FROM clause and the FROM NAMED clause.
- The FROM and FROM NAMED keywords allow a query to specify an RDF dataset by reference:
 - A default graph consisting of the RDF merge of the graphs referred to in the FROM clauses, and
 - A set of named graph pairs, one from each FROM NAMED clause
 - If there is no FROM clause, but one or more FROM NAMED clauses, then the dataset includes an empty graph for the default graph.

RDF Dataset- The Relationship between Named and Default Graphs



Default graph

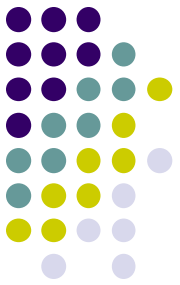
```
@PREFIX dc: <http://purl.org/dc/elements/1.1/> .  
  
<http://example.org/bob> dc:publisher "Bob" .  
<http://example.org/alice> dc:publisher "Alice" .
```

Named graph: http://example.org/bob

```
@PREFIX foaf: <http://xmlns.com/foaf/0.1> .  
  
_:a foaf:name "Bob" .  
_:a foaf:mbox <mailto:bob@oldcorp.example.org> .
```

Named graph: http://example.org/alice

```
@PREFIX foaf: <http://xmlns.com/foaf/0.1> .  
  
_:a foaf:name "Alice" .  
_:a foaf:mbox <mailto:alice@work.example.org> .
```



FROM

- Data (stored at <http://example.org/foaf/aliceFoaf>)

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

Default Graph

```
_:a foaf:name "Alice" .
```

```
_:a foaf:mbox <mailto:alice@work.example> .
```

- Query

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

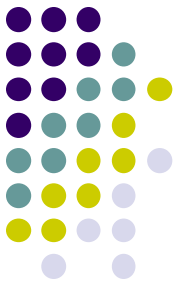
```
SELECT ?name
```

```
FROM <http://example.org/foaf/aliceFoaf>
```

```
WHERE { ?x foaf:name ?name }
```

- Result

name
"Alice"



FROM NAMED

- Data (named graphs)

```
# Named Graph: http://example.org/bob
@PREFIX foaf:      <http://xmlns.com/foaf/0.1/> .

_:a foaf:name      "Bob" .
_:a foaf:mbox       <mailto:bob@oldcorp.example.org> .
```

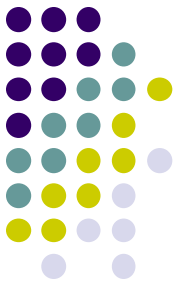
```
# Named Graph: http://example.org/alice
@PREFIX foaf:      <http://xmlns.com/foaf/0.1/> .

_:a foaf:name      "Alice" .
_:a foaf:mbox       <mailto:alice@work.example> .
```

- Query

```
PREFIX foaf:      <http://xmlns.com/foaf/0.1/> .
SELECT ?name
FROM NAMED        <http://example.org/alice>
FROM NAMED        <http://example.org/bob>
WHERE { ?x foaf:name ?name }
```

name
"Alice"
"Bob"



FROM and FROM NAMED

- Data

```
# Default Graph: http://example.org/dft.ttl
@PREFIX dc:      <http://purl.org/dc/elements/1.1/> .

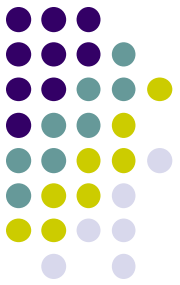
<http://example.org/bob>  dc:publisher      "Bob Hacker" .
<http://example.org/alice> dc:publisher      "Alice Hacker" .
```

```
# Named Graph: http://example.org/bob
@PREFIX foaf:    <http://xmlns.com/foaf/0.1/> .

_:a foaf:name      "Bob" .
_:a foaf:mbox       <mailto:bob@oldcorp.example.org> .
```

```
# Named Graph: http://example.org/alice
@PREFIX foaf:    <http://xmlns.com/foaf/0.1/> .

_:a foaf:name      "Alice" .
_:a foaf:mbox       <mailto:alice@work.example> .
```



FROM and FROM NAMED

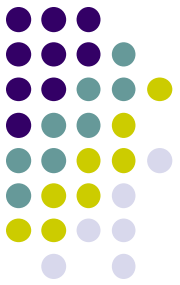
- Query

```
PREFIX foaf: <http://xmlns.com/foaf/1.0/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?who ?g ?mbox
FROM <http://example.org/dft.ttl>
FROM NAMED <http://example.org/alice>
FROM NAMED <http://example.org/bob>
WHERE
{
    ?g dc:publisher ?who .
    GRAPH ?g { ?x foaf:mbox ?mbox }
}
```

- Result

who	g	mbox
"Bob Hacker"	http://example.org/bob	<mailto:bob@oldcorp.example.org>
"Alice Hacker"	http://example.org/alice	<mailto:alice@work.example.org>



GRAPH

- GRAPH is used to match patterns against named graphs.
- Data

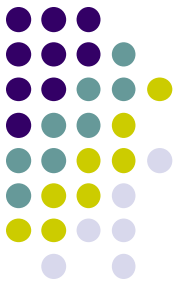
```
# Named Graph: http://example.org/foaf/aliceFoaf
@PREFIX foaf:    <http://xmlns.com/foaf/0.1/> .
@PREFIX rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@PREFIX rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .

_:a foaf:name      "Alice" .
_:a foaf:mbox      <mailto:alice@work.example> .
_:a foaf:knows     _:b .

_:b foaf:name      "Bob" .
_:b foaf:mbox      <mailto:bob@work.example> .
_:b foaf:nick      "Bobby" .
_:b foaf:seeAlso   <http://example.org/foaf/bobFoaf> .

<http://example.org/foaf/bobFoaf> rdf:type foaf:PersonalProfileDocument .
```

GRAPH



- Data

```
# Named Graph: http://example.org/foaf/bobFoaf
@PREFIX foaf:      <http://xmlns.com/foaf/0.1/> .
@PREFIX rdf:       <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@PREFIX rdfs:      <http://www.w3.org/2000/01/rdf-schema#> .

_:z foaf:mbox      <mailto:bob@work.example> .
_:z foaf:seeAlso   <http://example.org/foaf/bobFoaf> .
_:z foaf:nick      "Robert" .

<http://example.org/foaf/bobFoaf> rdf:type foaf:PersonalProfileDocument .
```



GRAPH – accessing graph names

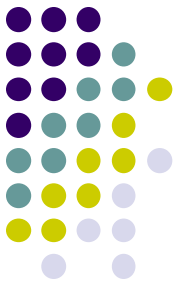
- src variable bounds to URIs of the graph being matched
- Query

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

SELECT ?src ?bobNick
FROM NAMED <http://example.org/foaf/aliceFoaf>
FROM NAMED <http://example.org/foaf/bobFoaf>
WHERE
{
  GRAPH ?src
  { ?x foaf:mbox <mailto:bob@work.example> .
    ?x foaf:nick ?bobNick
  }
}
```

- Result

src	bobNick
<http://example.org/foaf/aliceFoaf>	“Bobby”
<http://example.org/foaf/bobFoaf>	“Robert”



Restricting by Graph URI

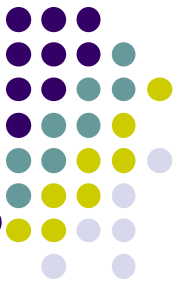
- The query can restrict the matching applied to a specific graph by supplying the graph URI.
- Query

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

SELECT ?nick
FROM NAMED <http://example.org/foaf/aliceFoaf>
FROM NAMED <http://example.org/foaf/bobFoaf>
WHERE
{
  GRAPH data:bobFoaf
  { ?x foaf:mbox <mailto:bob@work.example> .
    ?x foaf:nick ?Nick
  }
}
```

- Result

nick
"Robert"



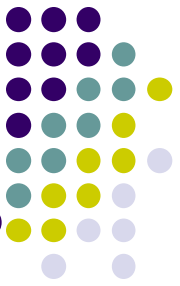
Restricting possible Graph URIs

- A variable used in the GRAPH clause may also be used in another GRAPH clause or in a graph pattern.
- Query

```
PREFIX data:      <http://example.org/foaf/>
PREFIX foaf:      <http://xmlns.com/foaf/0.1/>
PREFIX rdfs:      <http://www.w3.org/2000/01/rdf-schema#>
```

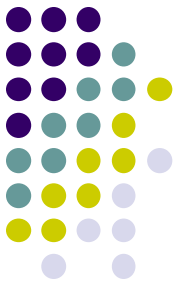
```
SELECT ?mbox ?nick ?ppd
FROM NAMED <http://example.org/foaf/aliceFoaf>
FROM NAMED <http://example.org/foaf/bobFoaf>
WHERE
{
  GRAPH data:aliceFoaf
  { ?alice foaf:mbox <mailto:alice@work.example> ;
    foaf:knows ?whom .
    ?whom foaf:mbox ?mbox ;
    rdfs:seeAlso ?ppd .
    ?ppd a foaf:PersonProfileDocument . } .
  GRAPH ?ppd
  { ?x foaf:mbox ?mbox ;
    foaf:nick ?Nick
  }
}
```

Restricting possible Graph URIs



- Result

mbox	nick	ppd
<mailto:bob@work.example>	"Robert"	<http://example.org/foaf/bobFoaf>



Named and Default Graphs

● Data

Default graph

```
@PREFIX dc:      <http://purl.org/dc/elements/1.1/> .
@PREFIX g:       <tag:example.org, 2005-06-06:> .
@PREFIX xsd:     <http://www.w3.org/2001/XMLSchema#> .
```

```
g:graph1 dc:publisher      "Bob" .
g:graph1 dc:date "2004-12-06"^^xsd:date .
```

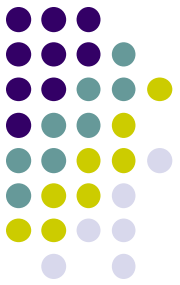
```
g:graph2 dc:publisher      "Bob" .
g:graph2 dc:date "2005-01-10"^^xsd:date .
```

Graph: locally allocated IRI: tag:example.org, 2004-12-06:graph1

```
@PREFIX foaf:    <http://xmlns.ocm/foaf/0.1/> .
```

```
_:a foaf:name      "Alice" .
_:a foaf:mbox      <mailto:alice@work.example> .
```

```
_:b foaf:name      "Bob" .
_:b foaf:mbox      <mailto:bob@oldcorp.example.org> .
```



Named and Default Graphs

- Data

```
# Graph: locally allocated IRI: tag:example.org, 2005-06-06:graph2
@PREFIX foaf:      <http://xmlns.ocm/foaf/0.1/> .

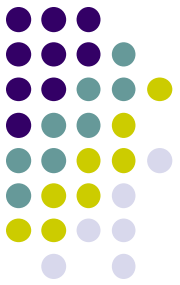
_:a foaf:name      "Alice" .
_:a foaf:mbox      <mailto:alice@work.example> .

_:b foaf:name      "Bob" .
_:b foaf:mbox      <mailto:bob@newcorp.example.org> .
```

- Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1>
PREFIX dc:   <http://purl.org/dc/elements/1.1>

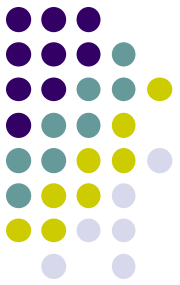
SELECT ?name ?mbox ?date
WHERE
  { ?g dc:publisher ?name ;
        dc:date ?date .
    GRAPH ?g
      { ?person foaf:name ?name ; foaf:mbox ?mbox }
```



Named and Default Graphs

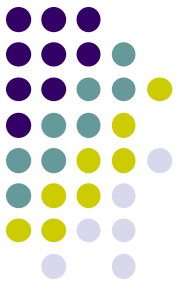
- Result

name	mbox	date
"Bob"	<mailto:bob@oldcorp.example.org>	"2004-12-06"^^xsd:date
"Bob"	<mailto:bob@newcorp.example.org>	"2005-06-06"^^xsd:date



Query Ordering

- A solution sequence modifier is one of
 - Order: put the solutions in order
 - Projection: choose certain variables
 - Distinct: ensure solutions in the sequence are unique
 - Reduced: permit elimination of some non-unique solutions
 - Offset: control where the solutions start from in the overall sequence of solutions
 - Limit: restrict the number of solutions



ORDER BY

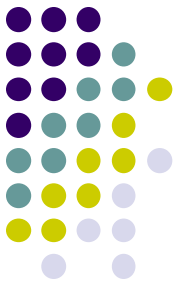
- ASC() – ascending
- DESC() - descending

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

```
SELECT ?name  
WHERE { ?x foaf:name ?name }  
ORDER BY ?name
```

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

```
SELECT ?name  
WHERE { ?x foaf:name ?name }  
ORDER BY DESC(?name)
```



Projection

- Projection is reflected in SELECT query form by choosing certain variables

```
@Prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

```
_:a foaf:name      "Alice" .
```

```
_:a foaf:mbox      <mailto:alice@work.example> .
```

```
_:b foaf:name      "Bob" .
```

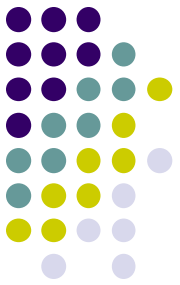
```
_:b foaf:mbox      <mailto:bob@work.example> .
```

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

```
SELECT ?name
```

```
WHERE { ?x foaf:name ?name }
```

name
"Bob"
"Alice"



Duplicate solutions

- Using DISTINCT or REDUCED to deal with duplicate solutions

```
@Prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:x foaf:name      "Alice" .
_:x foaf:mbox      <mailto:alice@example.com> .

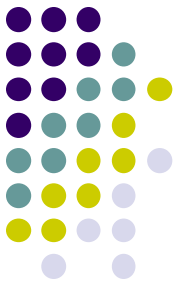
_:y foaf:name      "Alice" .
_:y foaf:mbox      <mailto:asmith@example.com> .

_:z foaf:name      "Alice" .
_:z foaf:mbox      <mailto:alice.smith@example.com> .
```

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

```
SELECT ?name
WHERE { ?x foaf:name ?name }
```

name
"Alice"
"Alice"
"Alice"



Duplicate solutions

- Using DISTINCT or REDUCED to deal with duplicate solutions

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

```
SELECT DISTINCT ?name  
WHERE { ?x foaf:name ?name }
```

name
"Alice"

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

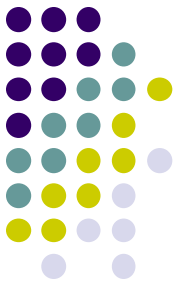
```
SELECT REDUCED ?name  
WHERE { ?x foaf:name ?name }
```

name
"Alice"

name
"Alice"
"Alice"

name
"Alice"
"Alice"
"Alice"

Not recommended



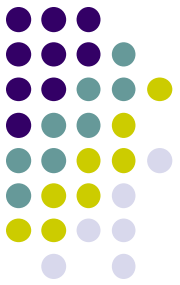
OFFSET

- OFFSET tells from which number the solutions should be generated from
- LIMIT sets an upper bound on the number of solutions returned.

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

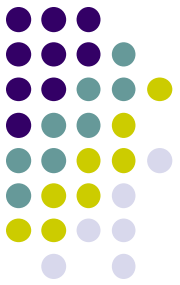
```
SELECT ?name  
WHERE { ?x foaf:name ?name }  
ORDER BY ?name  
LIMIT 5  
OFFSET 10
```

Returned Solutions are generated from the no. 11th solutions with no more than 5 solutions in total



Query Forms

- SPARQL has four query forms
 - SELECT: return all, or a subset of, the variable bound in a query pattern match
 - CONSTRUCT: returns an RDF graph constructed by substituting variables in a set of triple templates
 - ASK: returns a boolean indicating whether a query pattern matches or not
 - DESCRIBE: returns an RDF graph that describes the resources found



SELECT

• Data

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

```
_:a foaf:name      "Alice" .
```

```
_:a foaf:knows    _:b .
```

```
_:a foaf:knows    _:c .
```

```
_:b foaf:name      "Bob" .
```

```
_:c foaf:name      "Clare" .
```

```
_:c foaf:nick      "CT" .
```

• Query

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

```
SELECT ?nameX ?nameY ?nickY
```

```
WHERE
```

```
{ ?x foaf:knows ?y ;
```

```
    foaf:name ?nameX .
```

```
  ?y foaf:name ?nameY .
```

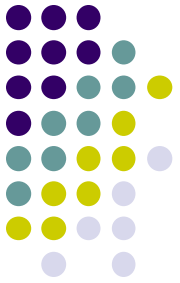
```
  OPTIONAL { ?y foaf:nick ?nickY }
```

```
}
```

SELECT

- Result

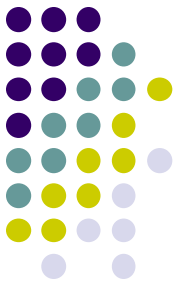
nameX	nameY	nickY
"Alice"	"Bob"	
"Alice"	"Clare"	"CT"



SELECT

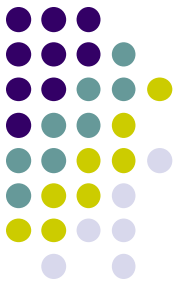
- Result in SPARQL Query Results XML Format

```
<?xml version="1.0" ?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="nameX"/>
    <variable name="nameY"/>
    <variable name="nickY"/>
  </head>
  <results>
    <result>
      <binding name="nameX">
        <literal>Alice</literal>
      </binding>
      <binding name="nameY">
        <literal>Bob</literal>
      </binding>
    </result>
    <result>
      <binding name="nameX">
        <literal>Alice</literal>
      </binding>
      <binding name="nameY">
        <literal>Clare</literal>
      </binding>
      <binding name="nickY">
        <literal>CT</literal>
      </binding>
    </result>
  </results>
</sparql>
```



References:

- SPARQL Query Language
<http://www.w3.org/TR/rdf-sparql-query/>
- SPARQL Protocol:
<http://www.w3.org/TR/rdf-sparql-protocol/>
- SPARQL Query Results XML Format:
<http://www.w3.org/TR/rdf-sparql-XMLres/>



Fun Examples

- [http://www.cambridgesemantics.com/2008/09/sparql-by-example/#\(1\)](http://www.cambridgesemantics.com/2008/09/sparql-by-example/#(1))