18A – INTRODUCTION TO RDF AND SPARQL

CS 1656

Introduction to Data Science

Alexandros Labrinidis – http://labrinidis.cs.pitt.edu
University of Pittsburgh

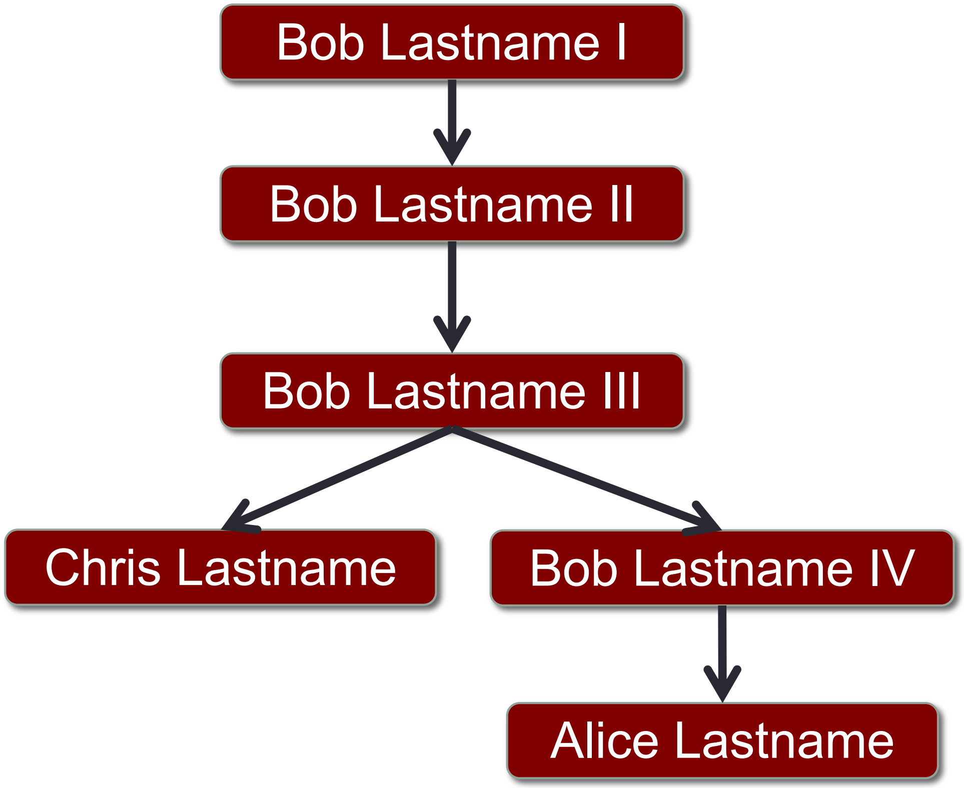
INTRODUCTION TO RDF

Data Models so far

- **Relational Model / SQL**
  - Organizes data in table form
  - Strict format
  - Related data are in one tuple
  - Tries to minimize redundancy
  - Relationships are captured with tables as well
  - Often combine data from multiple tables (through joins)
- **XML**
  - Organizes data in documents
  - Flexible format
  - Related data are in one element
  - Data organized hierarchically
  - Sometimes data combined from multiple documents

What about graph data?

- Imagine a simple genealogy data set



Graph data in the relational model

- In the relational model, you could store as a table
- **How?**

`Genealogy`

name	childOF	birthyear
Alice Lastname	Bob Lastname IV	2010
Bob Lastname IV	Bob Lastname III	1980
Chris Lastname	Bob Lastname III	1982
Bob Lastname III	Bob Lastname II	1966
Bob Lastname II	Bob Lastname I	1934
Bob Lastname I		1910

- How would you find in SQL all descendants of Bob Lastname I?

Querying graph data in SQL

- **Find all children of Bob Lastname I:**

```
select name from genealogy
where childOF = “Bob Lastname I”
```

- **Find all grandchildren of Bob Lastname I:**

```
select g1.name
from genealogy as g1, genealogy as g2
where g1.childOF = g2.name
  and g2.childOF = “Bob Lastname I”
```

- All great-grandchildren requires three copies of genealogy relation, etc.

An idea

- *What if instead of storing the entire structure, we just stored the individual connections?*
- *And made queries connection-centric?*

Storing graph info in triplets

General form:

- Subject (resource identifier)
  - E.g., Bob Lastname III
- Predicate (property name)
  - E.g., is-a-child-of
- Object (property value)
  - E.g., Bob Lastname II

Example

[Source: http://learningsparql.com]

```
# filename: ex002.ttl

@prefix ab: <http://learningsparql.com/ns/addressbook#> .

ab:richard ab:homeTel "(229) 276-5135" .
ab:richard ab:email   "richard49@hotmail.com" .

ab:cindy ab:homeTel "(245) 646-5488" .
ab:cindy ab:email   "cindym@gmail.com" .

ab:craig ab:homeTel "(194) 966-1505" .
ab:craig ab:email   "craigellis@yahoo.com" .
ab:craig ab:email   "c.ellis@usairwaysgroup.com" .
```

INTRODUCTION TO SPARQL

SPARQL by Example – ex003.rq

[Source: http://learningsparql.com]

```
# filename: ex003.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?craigEmail
WHERE
{ ab:craig ab:email ?craigEmail . }
```

RESULT:

craigEmail
“c.ellis@usairwaysgroup.com”
“craigellis@yahoo.com”

SPARQL by Example – ex007.rq

```
# filename: ex007.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?craigEmail FROM <ex002.ttl>
WHERE
{ ab:craig ab:email ?craigEmail . }
```

SPARQL by Example – ex010.rq

```
# filename: ex010.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?propertyName ?propertyValue
WHERE
{ ab:cindy ?propertyName ?propertyValue . }
```

RESULT:

propertyName	propertyValue
ab:email	“cindym@gmail.com”
ab:homeTel	"(245) 646-5488”

SPARQL by Example – ex012.ttl

```
# filename: ex012.ttl

@prefix ab: <http://learningsparql.com/ns/addressbook#> .
@prefix d:  <http://learningsparql.com/ns/data#> .

d:i0432 ab:firstName "Richard" .
d:i0432 ab:lastName  "Mutt" .
d:i0432 ab:homeTel   "(229) 276-5135" .
d:i0432 ab:email     "richard49@hotmail.com" .

d:i9771 ab:firstName "Cindy" .
d:i9771 ab:lastName  "Marshall" .
d:i9771 ab:homeTel   "(245) 646-5488" .
d:i9771 ab:email     "cindym@gmail.com" .

d:i8301 ab:firstName "Craig" .
d:i8301 ab:lastName  "Ellis" .
d:i8301 ab:email     "craigellis@yahoo.com" .
d:i8301 ab:email     "c.ellis@usairwaysgroup.com" .
```

Graph representation?

SPARQL by Example – ex013.rq

```
# filename: ex013.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?craigEmail
WHERE
{
  ?person ab:firstName "Craig" .
  ?person ab:email ?craigEmail .
}
```

craigEmail
“c.ellis@usairwaysgroup.com”
“craigellis@yahoo.com”

SPARQL by Example – ex015.rq

```
# filename: ex015.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?craigEmail
WHERE
{
  ?person ab:firstName "Craig" .
  ?person ab:lastName  "Ellis" .
  ?person ab:email ?craigEmail .
}
```

craigEmail
“c.ellis@usairwaysgroup.com”
“craigellis@yahoo.com”

SPARQL by Example – ex017.rq

```
# filename: ex017.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?first ?last
WHERE
{
  ?person ab:homeTel "(229) 276-5135" .
  ?person ab:firstName ?first .
  ?person ab:lastName  ?last .
}
```

first	last
Richard	Mutt

SPARQL by Example – ex019.rq

```
# filename: ex019.rq

PREFIX a: <http://learningsparql.com/ns/addressbook#>

SELECT ?propertyName ?propertyValue

WHERE
{
  ?person a:firstName "Cindy" .
  ?person a:lastName  "Marshall" .
  ?person ?propertyName ?propertyValue .
}
```

propertyName	propertyValue
a:mail	"cindym@gmail.com"
a:homeTel	"(245) 646-5488"
a:lastName	"Marshall"
a:firstName	"Cindy"

SPARQL by Example – ex021.rq

```
# filename: ex021.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT *
WHERE
{
  ?s ?p ?o .
  FILTER (regex(?o, "yahoo","i"))
}
```

S	P	O
<http://learningsparql.com/ns/data#i8301>	ab:email	"craigellis@yahoo.com"

SPARQL by Example – ex023.rq

```
# filename: ex023.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?craigEmail ?homeTel
WHERE
{
  ?person ab:firstName "Craig" .
  ?person ab:lastName  "Ellis" .
  ?person ab:email ?craigEmail .
  ?person ab:homeTel ?homeTel .
}
```

craigEmail	homeTel

Why empty?